



The Weblate Manual

Release 4.16

Michal Čihař

01.03.2023

1	Dokumentation für Benutzer	1
1.1	Weblate-Grundlagen	1
1.2	Registrierung und Benutzerprofil	1
1.3	Mit Weblate übersetzen	10
1.4	Herunterladen und Hochladen von Übersetzungen	20
1.5	Glossar	23
1.6	Überprüfungen und Korrekturen	26
1.7	Suche	59
1.8	Übersetzungsabläufe	64
1.9	Häufig gestellte Fragen	68
1.10	Unterstützte Dateiformate	76
1.11	Integration der Versionsverwaltung	99
1.12	Weblates REST API	106
1.13	Weblate-Client	155
1.14	Weblates Python API	161
2	Dokumentation für Administratoren	163
2.1	Konfigurationsanweisungen	163
2.2	Weblate-Bereitstellungen	232
2.3	Upgrade von Weblate	233
2.4	Sichern und Verschieben von Weblate	241
2.5	Authentifizierung	249
2.6	Zugriffssteuerung	259
2.7	Übersetzungsprojekte	269
2.8	Sprachdefinitionen	287
2.9	Kontinuierliche Lokalisierung	291
2.10	Licensing translations	299
2.11	Übersetzungsrichtlinien	301
2.12	Überprüfungen und Korrekturen	307
2.13	Automatische Vorschläge konfigurieren	318
2.14	Erweiterungen	329
2.15	Übersetzungsspeicher	347
2.16	Konfiguration	349
2.17	Sample configuration	376
2.18	Management commands	392
2.19	Ankündigungen	404
2.20	Komponentenlisten	406
2.21	Optionale Weblate-Module	407
2.22	Anpassen von Weblate	412
2.23	Verwaltungsoberfläche	415
2.24	Getting support for Weblate	423

2.25	Rechtliche Dokumente	426
3	Dokumentation für Mitwirkende	428
3.1	Zu Weblate beitragen	428
3.2	Code für Weblate beisteuern	430
3.3	Weblate-Quellcode	434
3.4	Fehlersuche in Weblate	435
3.5	Weblate-Internia	437
3.6	Erweiterungen entwickeln	438
3.7	Weblate-Frontend	440
3.8	Probleme in Weblate melden	441
3.9	Weblate-Testsuite und kontinuierliche Integration	442
3.10	Datenschemata	443
3.11	Weblate-Veröffentlichungen	447
3.12	Sicherheit und Datenschutz	448
3.13	Mitarbeit an Weblate-Modulen	449
3.14	Über Weblate	450
3.15	Lizenz	451
4	Änderungshistorie	452
4.1	Weblate 4.16	452
4.2	Weblate 4.15.2	452
4.3	Weblate 4.15.1	453
4.4	Weblate 4.15	453
4.5	Weblate 4.14.2	454
4.6	Weblate 4.14.1	454
4.7	Weblate 4.14	455
4.8	Weblate 4.13.1	455
4.9	Weblate 4.13	455
4.10	Weblate 4.12.2	456
4.11	Weblate 4.12.1	456
4.12	Weblate 4.12	457
4.13	Weblate 4.11.2	457
4.14	Weblate 4.11.1	457
4.15	Weblate 4.11	458
4.16	Weblate 4.10.1	458
4.17	Weblate 4.10	459
4.18	Weblate 4.9.1	459
4.19	Weblate 4.9	459
4.20	Weblate 4.8.1	460
4.21	Weblate 4.8	461
4.22	Weblate 4.7.2	461
4.23	Weblate 4.7.1	461
4.24	Weblate 4.7	462
4.25	Weblate 4.6.2	462
4.26	Weblate 4.6.1	463
4.27	Weblate 4.6	463
4.28	Weblate 4.5.3	464
4.29	Weblate 4.5.2	464
4.30	Weblate 4.5.1	464
4.31	Weblate 4.5	465
4.32	Weblate 4.4.2	465
4.33	Weblate 4.4.1	466
4.34	Weblate 4.4	466
4.35	Weblate 4.3.2	467
4.36	Weblate 4.3.1	467
4.37	Weblate 4.3	468
4.38	Weblate 4.2.2	469

4.39	Weblate 4.2.1	469
4.40	Weblate 4.2	469
4.41	Weblate 4.1.1	470
4.42	Weblate 4.1	470
4.43	Weblate 4.0.4	471
4.44	Weblate 4.0.3	472
4.45	Weblate 4.0.2	472
4.46	Weblate 4.0.1	472
4.47	Weblate 4.0	473
4.48	Weblate 3.x-Reihe	473
4.49	Weblate 2.x-Reihe	485
4.50	Weblate 1.x-Reihe	496
4.51	Weblate 0.x-Reihe	501
Python-Modulindex		504
HTTP Routing Table		505
Stichwortverzeichnis		508

1.1 Weblate-Grundlagen

1.1.1 Projekt- und Komponentenstruktur

In Weblate sind die Übersetzungen in Projekten und Komponenten organisiert. Jedes Projekt kann eine Reihe von Komponenten enthalten, die Übersetzungen in einzelne Sprachen enthalten. Die Komponente entspricht einer übersetzbaren Datei (zum Beispiel *GNU gettext* oder *Android String-Ressourcen*). Die Projekte helfen Ihnen, die Komponenten in logischen Gruppen zu organisieren (z. B. um alle in einer Anwendung verwendeten Übersetzungen zu gruppieren).

Intern verfügt jedes Projekt über Übersetzungen in gemeinsamen Zeichenketten, die standardmäßig auf andere Komponenten des Projekts übertragen werden. Dies erleichtert die Last der sich wiederholenden und mehrere Versionen umfassenden Übersetzung. Die Weitergabe von Übersetzungen kann für jede *Komponentenkonfiguration* mit *Verbreitung von Übersetzungen erlauben* deaktiviert werden, falls die Übersetzungen voneinander abweichen sollten.

Siehe auch:

`../devel/integration`

1.2 Registrierung und Benutzerprofil

1.2.1 Registrierung

Everybody can browse projects, view translations or suggest translations by default. Only registered users are allowed to actually save changes, and are credited for every translation made.

You can register by following a few simple steps:

1. Fill out the registration form with your credentials.
2. Activate registration by following the link in the e-mail you receive.
3. Optionally adjust your profile to choose which languages you know.

1.2.2 Übersichtsseite

When you sign in, you will see an overview of projects and components, as well as their respective translation progression.

Neu in Version 2.5.

Components of projects you are watching are shown by default, and cross-referenced with your preferred languages.

Hinweis: You can switch to different views using the navigation tabs.

The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo and menu items: Dashboard, Projects, Languages, and Checks. Below this is a user profile section with a 'Your profile' link and a row of tabs: Languages, Preferences (which is active), Notifications, Account, Profile, Teams, Licenses, Audit log, and API access. The main content area is titled 'Preferences' and contains several settings:

- ☐ Hide completed translations on the dashboard
- Translation editor mode**: A dropdown menu currently set to 'Full editor'.
- Zen editor mode**: A dropdown menu currently set to 'Top to bottom'.
- Number of nearby strings**: A text input field containing the number '15'. Below it is a small explanatory text: 'Number of nearby strings to show in each direction in the full editor.'
- ☒ Show secondary translations in the Zen mode
- ☐ Hide source if a secondary translation exists
- Editor link**: A text input field. Below it is a small explanatory text: 'Enter a custom URL to be used as link to the source code. You can use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders.'
- Special characters**: A text input field. Below it is a small explanatory text: 'You can specify additional special visual keyboard characters to be shown while translating. It can be useful for characters you use frequently, but are hard to type on your keyboard.'
- Default dashboard view**: Two radio buttons, with 'Watched translations' selected and 'Suggested translations' unselected.

At the bottom of the preferences panel is a 'Save' button.

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

The menu has these options:

- *Projects > Browse all projects* in the main menu showing translation status for each project on the Weblate instance.
- Selecting a language in the main menu *Languages* will show translation status of all projects, filtered by one of your primary languages.
- *Watched translations* in the Dashboard will show translation status of only those projects you are watching, filtered by your primary languages.

In addition, the drop-down can also show any number of *component lists*, sets of project components preconfigured by the Weblate administrator, see [Komponentenlisten](#).

You can configure your personal default dashboard view in the *Preferences* section of your user profile settings.

Bemerkung: When Weblate is configured for a single project using `SINGLE_PROJECT` in the `settings.py` file (see *Konfiguration*), the dashboard will not be shown, as the user will be redirected to a single project or component instead.

1.2.3 Benutzerprofil

The user profile is accessible by clicking your user icon in the top-right of the top menu, then the *Settings* menu.

The user profile contains your preferences. Name and e-mail address is used in VCS commits, so keep this info accurate.

Bemerkung: All language selections only offer currently translated languages.

Hinweis: Request or add other languages you want to translate by clicking the button to make them available too.

Sprachen

1.2.4 Sprache der Benutzeroberfläche

Wählen Sie die Sprache, in der die Benutzeroberfläche angezeigt werden soll.

Übersetzte Sprachen

Choose which languages you prefer to translate, and they will be offered on the main page of watched projects, so that you have easier access to these all translations in each of those languages.

Weblate

Dashboard

Projects

Languages

Checks

+

Dashboard

Watched translations13

Suggested translations5

Insights

Search

Translation	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
<div><div></div><div>WeblateOrg/Android — Czech</div><div><div></div><div></div><div>MIT</div></div></div>	76%	3	3	10			
<div><div></div><div>WeblateOrg/Django — Czech</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	96%	1	12	69	4		
<div><div></div><div>WeblateOrg/Django — Hebrew</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	92%	2	15	80			
<div><div></div><div>WeblateOrg/Django — Hungarian</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	69%	8	109	671	1		
<div><div></div><div>WeblateOrg/Djangojs — Hungarian</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	96%	2	6	28			
<div><div></div><div>WeblateOrg/Djangojs — Hebrew</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	✓						
<div><div></div><div>WeblateOrg/Djangojs — Czech</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	✓						
<div><div></div><div>WeblateOrg/Language names — Czech</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	✓						
<div><div></div><div>WeblateOrg/Language names — Hungarian</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	81%	4	5	32			
<div><div></div><div>WeblateOrg/Language names — Hebrew</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	✓						
<div><div></div><div>WeblateOrg/WebplateOrg — Hungarian</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	✓						
<div><div></div><div>WeblateOrg/WebplateOrg — Czech</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	✓						
<div><div></div><div>WeblateOrg/WebplateOrg — Hebrew</div><div><div></div><div></div><div>GPL-3.0</div></div></div>	✓						

Zweitsprachen

You can define which secondary languages are shown to you as a guide while translating. An example can be seen in the following image, where the Hebrew language is shown as secondarily:

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb 'WeblateOrg / Django / Czech / Translate' is visible. A progress bar indicates 'translated 96%'. The main area shows a translation for the string 'קבצים' (Files) from Hebrew to Czech. The Czech input field contains 'Soubory'. Below the input fields are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A sidebar on the right contains a 'Glossary' section with 'English' and 'Czech' tabs, and a 'String information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'. At the bottom, there's a table of 'Nearby strings' with columns for 'Language' and 'Target string'.

1.2.5 Einstellungen

Standardansicht der Übersichtsseite

On the *Preferences* tab, you can pick which of the available dashboard views to present by default. If you pick the *Component list*, you have to select which component list will be displayed from the *Default component list* drop-down.

Siehe auch:

Komponentenlisten

Editor-URL

A source code link is shown in the web-browser configured in the *Komponentenkonfiguration* by default.

Hinweis: By setting the *Editor link*, you use your local editor to open the VCS source code file of translated strings. You can use *Template markup*.

Usually something like `editor://open/?file={{filename}}&line={{line}}` is a good option.

Siehe auch:

You can find more info on registering custom URL protocols for the editor in the *Nette documentation*.

1.2. Registrierung und Benutzerprofil

Sonderzeichen

Additional special characters to include in the *Visuelle Tastatur*.

1.2.6 Benachrichtigungen

Subscribe to various notifications from the *Notifications* tab. Notifications for selected events on watched or administered projects will be sent to you per e-mail.


Some of the notifications are sent only for events in your languages (for example about new strings to translate), while some trigger at component level (for example merge errors). These two groups of notifications are visually separated in the settings.

You can toggle notifications for watched projects and administered projects and it can be further tweaked (or muted) per project and component. Visit the component overview page and select appropriate choice from the *Watching* menu.

In case *Automatically watch projects on contribution* is enabled you will automatically start watching projects upon translating a string. The default value depends on *DEFAULT_AUTO_WATCH*.

Bemerkung: You will not receive notifications for your own actions.

Hinweis: Sending out notifications is limited, you will not receive more than 1000 e-mails per day. Any further notifications for you will be discarded.

 Weblate
 Dashboard
 Projects
 Languages
 Checks

Your profile
 Languages
 Preferences
 Notifications
 Account
 Profile
 Teams
 Licenses
 Audit log
 API access

Watched projects
 ⓘ

☒ Automatically watch projects on contribution
 Whenever you translate a string in a project, you will start watching it.

Watched projects

Search...

Available:

WeblateOrg

 Chosen:

WeblateOrg

You can receive notifications for watched projects and they are shown on the dashboard by default.
 Add all projects you want to translate to see them as watched projects on the dashboard.

Save

Notification settings
 ⓘ

Other projects
 Watched projects
 Managed projects

Component wide notifications

You will receive a notification for every such event in your watched projects.

Repository failure	Do not notify
Repository operation	Do not notify
Component locking	Do not notify
Changed license	Do not notify
Parse error	Do not notify
Comment on own translation	Instant notification
Mentioned in comment	Instant notification
New language	Do not notify
New translation component	Do not notify
New announcement	Instant notification
New alert	Do not notify

Translation notifications

You will only receive these notifications for your translated languages in your watched projects.

New string	Do not notify
New contributor	Do not notify
New suggestion	Do not notify
New comment	Do not notify
Changed string	Do not notify
Translated string	Do not notify
Approved string	Do not notify
Pending suggestions	Do not notify
Unfinished strings	Do not notify

Save

Powered by Weblate 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

1.2.7 Benutzerkonto

Auf der Reiterkarte *Benutzerkonto* können Sie grundlegende Kontodetails einrichten, verschiedene Dienste verbinden, mit denen Sie sich bei Weblate anmelden können, Ihr Konto vollständig löschen oder Ihre Benutzerdaten herunterladen (siehe [Weblate Benutzerdatenexport](#)).

Bemerkung: Die Liste der Dienste hängt von Ihrer Weblate-Konfiguration ab, kann aber so gestaltet werden, dass sie beliebige Websites wie GitLab, GitHub, Google, Facebook, Bitbucket oder andere OAuth 2.0-Anbieter umfasst.

 Weblate
 Dashboard Projects Languages Checks
 + ⌵ ⋮

 Your profile
 Languages Preferences Notifications **Account** Profile Teams Licenses Audit log API access

Account

Username

Username may only contain letters, numbers or the following characters: @ . + - _

Full name


E-mail

You can add another e-mail address below.


Commit e-mail

Save

Current user identities

Identity	User ID	Action
 Password	testuser	Change password
 E-mail	weblate@example.org	Disconnect
 Google	weblate@example.org	Disconnect
 GitHub	123456	Disconnect
 Bitbucket	weblate	Disconnect

Add new association



E-mail

Removal

Account removal deletes all your private data.

Remove my account

User data

You can download all your private data.

Download user data

1.2.8 Benutzerprofil

Alle Felder auf dieser Seite sind optional und können jederzeit gelöscht werden. Indem Sie diese Felder ausfüllen, geben Sie uns Ihr Einverständnis, diese Daten überall dort weiterzugeben, wo Ihr Benutzerprofil erscheint.

Die Commit-E-Mail wird anstelle Ihrer Konto-E-Mail in Versionsverwaltungs-Commits verwendet. Verwenden Sie dies, um zu vermeiden, dass Ihre echte E-Mail dort durchsickert. Seien Sie sich bewusst, dass die Verwendung einer anderen E-Mail Ihre Beiträge auf anderen Servern trennen kann (zum Beispiel werden Ihre Beiträge nicht mehr mit Ihrem Profil auf GitHub verlinkt). Die private E-Mail kann mit `PRIVATE_COMMIT_EMAIL_OPT_IN` plattformweit aktiviert werden.

Für jeden Benutzer kann ein Avatar angezeigt werden (abhängig von `ENABLE_AVATARS`). Diese Bilder werden über <https://gravatar.com/> bezogen.

1.2.9 Lizenzen

1.2.10 API-Zugang

Sie können Ihr API-Zugangs-Token hier abrufen oder zurücksetzen.

1.2.11 Audit-Protokoll

Das Audit-Protokoll zeichnet die mit Ihrem Benutzerkonto durchgeführten Aktionen auf. Es protokolliert IP-Adresse und Browser für jede wichtige Aktion mit Ihrem Benutzerkonto. Die kritischen Aktionen lösen auch eine Benachrichtigung an eine primäre E-Mail-Adresse aus.

Siehe auch:

Hinter einem Reverse-Proxy ausführen

1.3 Mit Weblate übersetzen

Vielen Dank für Ihr Interesse, mit Weblate zu übersetzen. Projekte können entweder für die direkte Übersetzung eingerichtet werden oder durch die Annahme von Vorschlägen von Benutzern ohne Konto.

Es gibt zwei verschiedene Übersetzungsmodi:

- Das Projekt akzeptiert unmittelbare Übersetzungen
- Das Projekt akzeptiert nur Vorschläge, die automatisch für gültig erklärt werden, sobald eine definierte Anzahl an Stimmen abgegeben wurden

Weitere Informationen zum Übersetzungsablauf finden Sie unter *Übersetzungsabläufe*.

Optionen der Projektsichtbarkeit:

- öffentlich sichtbar
- nur für eine bestimmte Gruppe sichtbar

Siehe auch:

Zugriffssteuerung, Übersetzungsabläufe

1.3.1 Übersetzungsprojekte

Übersetzungsprojekte enthalten zugehörige Komponenten; Ressourcen für dieselbe Software, Buch oder Projekt.

Webate Dashboard Projects Languages Checks

WebateOrg translated 85%

Components Languages Info Search Insights Files Tools Manage Share Not watching

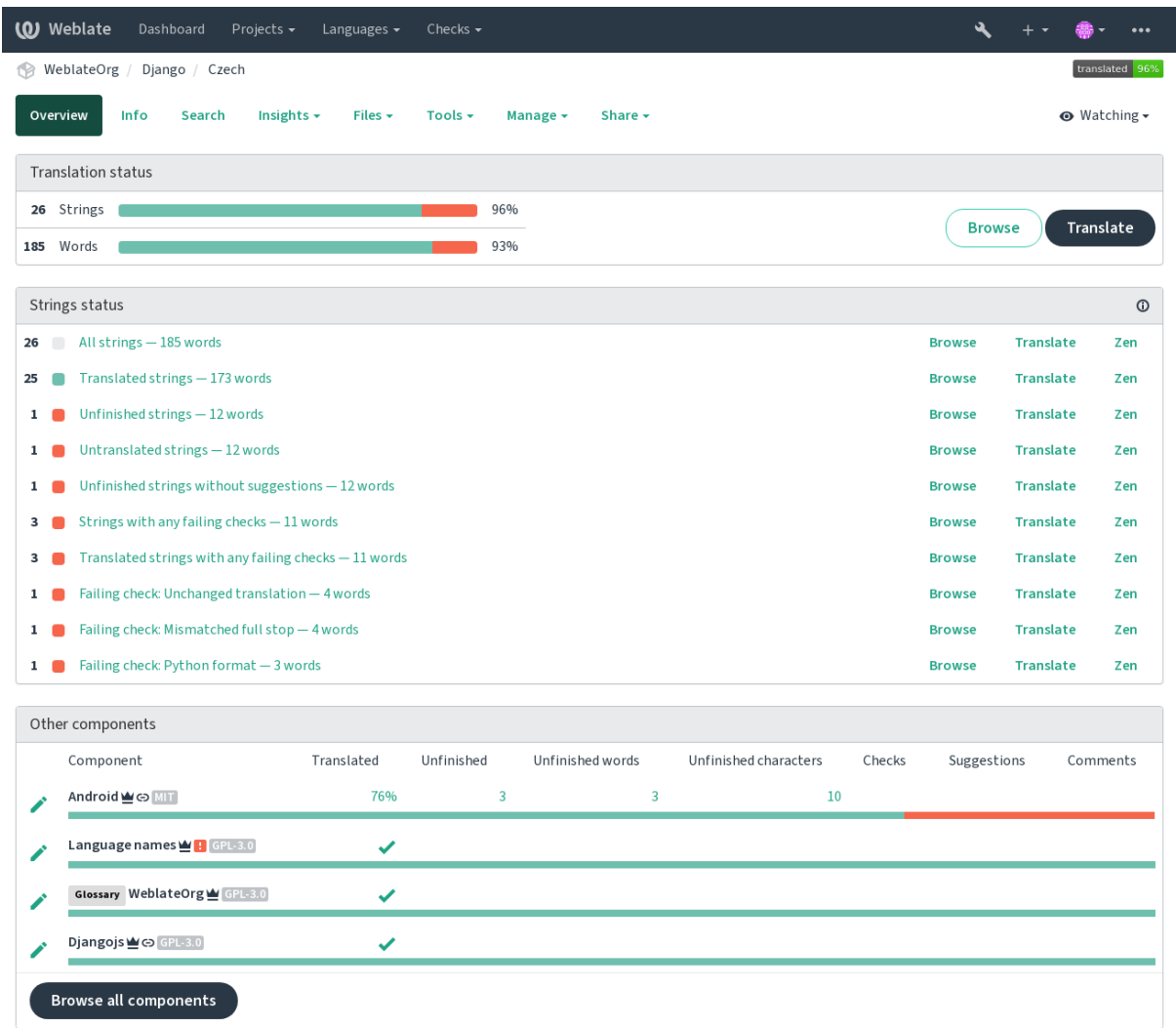
Component	Translated	Unfinished	Unfinished words	Checks	Suggestions	Comments
Android	79%	30	30	3		
Language names	95%	4	5			
Glossary WeblateOrg	100%	0	0	0		

Add new translation component

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

1.3.2 Übersetzungslinks

Nachdem man zu einer Komponente navigiert hat, führt eine Reihe von Links zur eigentlichen Übersetzung. Die Übersetzung ist weiter unterteilt in einzelne Qualitätsprüfungen, wie *Nicht übersetzte Zeichenketten* oder *Unvollständige Zeichenketten*. Wenn das gesamte Projekt fehlerfrei übersetzt wurde, ist *Alle Zeichenketten* weiterhin verfügbar. Alternativ können Sie auch das Suchfeld verwenden, um eine bestimmte Zeichenkette oder einen bestimmten Begriff zu finden.



Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.3.3 Vorschläge

Bemerkung: Die tatsächlichen Berechtigungen können je nach Konfiguration von Ihrer Weblate-Installation variieren.

Anonyme Benutzer können (standardmäßig) nur Übersetzungsvorschläge übermitteln. Dies steht angemeldeten Benutzern weiterhin zur Verfügung, falls Unklarheiten über die Übersetzung auftauchen, um zu veranlassen, dass andere Übersetzer sie überprüfen.

Die Übersetzungsvorschläge werden täglich gescannt, um Duplikate und Vorschläge zu entfernen, die der aktuellen Übersetzung entsprechen.

1.3.4 Kommentare

Es können drei Arten von Kommentaren erstellt werden: für Übersetzungen, Ausgangszeichenketten oder um Fehler im Quellcode zu melden, wenn diese Funktion mit *Quellenüberprüfung aktivieren* eingeschaltet wurde. Wählen Sie das passende Thema aus, welches Sie diskutieren möchten. Ausgangszeichenkette-Kommentare sind in jedem Fall gut geeignet, um Rückmeldung zum Quellcode zu geben, zum Beispiel, dass er umformuliert werden sollte oder um Fragen dazu zu stellen.

Sie können die Markdown-Syntax in allen Kommentaren verwenden und andere Benutzer mit `@mention` erwähnen.

Siehe auch:

report-source, *Überprüfungen der Ausgangszeichenketten*, *Quellenüberprüfung aktivieren*

1.3.5 Varianten

Varianten werden verwendet, um verschiedene Längenvarianten der Zeichenkette zu gruppieren. Das Frontend Ihres Projekts kann dann je nach Bildschirm- oder Fenstergröße unterschiedliche Zeichenketten verwenden.

Siehe auch:

variants, *Varianten*

1.3.6 Labels

Labels werden verwendet, um Zeichenketten innerhalb eines Projekts zu kategorisieren und den Lokalisierungsprozess weiter anzupassen (z. B. um Kategorien von Zeichenketten zu definieren).

Folgende Labels werden von Weblate verwendet:

Automatisch übersetzt

Zeichenkette wurde mit *Automatische Übersetzung* übersetzt.

Ausgangszeichenkette muss überprüft werden

Zeichenkette wurde mittels *Überprüfungen der Ausgangszeichenketten* zur Überprüfung markiert.

Siehe auch:

labels

1.3.7 Translating

Auf der Übersetzungsseite werden die Ausgangszeichenkette und ein Bearbeitungsbereich für ihre Übersetzung angezeigt. Handelt es sich bei der Übersetzung um eine Pluralform, werden mehrere Ausgangszeichenketten und Bearbeitungsbereiche angezeigt, die jeweils mit der Anzahl der Pluralformen der übersetzten Sprache beschrieben und beschriftet sind.

Alle besonderen Leerzeichen sind rot unterstrichen und durch graue Symbole gekennzeichnet. Mehr als ein nachfolgendes Leerzeichen ist ebenfalls rot unterstrichen, um den Übersetzer auf ein mögliches Formatierungsproblem hinzuweisen.

Auf dieser Seite können verschiedene zusätzliche Informationen angezeigt werden, von denen die meisten aus dem Quellcode des Projekts stammen (wie Kontext, Kommentare oder wo die Nachricht verwendet wird). Übersetzungsfelder für Zweitsprachen, die der Übersetzer in den Einstellungen ausgewählt hat, werden über der Ausgangszeichenkette angezeigt (siehe *Zweitsprachen*).

Unterhalb der Übersetzung finden die Übersetzer Vorschläge von anderen, die sie annehmen (✓), mit Änderungen annehmen (⇒) oder löschen (🗑️) können.

Plural

Wörter, die aufgrund ihrer numerischen Bezeichnung ihre Form ändern, werden Pluralformen genannt. Jede Sprache hat ihre eigene Definition von Pluralformen. Im Englischen zum Beispiel gibt es eine Pluralform. In der Singular-Definition von z. B. „car“ ist implizit ein Auto gemeint, in der Plural-Definition „cars“ sind zwei oder mehr Autos gemeint (oder das Konzept von Autos als Substantiv). Sprachen wie z. B. Tschechisch oder Arabisch haben mehr Pluralformen und auch ihre Regeln für Pluralformen sind anders.

Weblate bietet volle Unterstützung für jede dieser Formen in der jeweiligen Sprache (indem jeder Plural separat übersetzt wird). Die Anzahl der Felder und die Art der Verwendung in der übersetzten Anwendung oder dem Projekt hängt von der konfigurierten Pluralformel ab. Weblate zeigt die grundlegenden Informationen, und die [Language Plural Rules](#) des Unicode-Konsortium ist eine ausführlichere Beschreibung.

Siehe auch:

[Pluralformel](#)

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation task for the string '%(count)s word'. The interface is divided into several sections:

- Translation Panel:** Contains input fields for different plural forms. The English string is '%(count)s word'. The Czech translations are:
 - Czech, One:** '%(count)s slovo' (15/140 - 14)
 - Czech, Few:** '%(count)s slova' (15/140 - 15)
 - Czech, Many:** '%(count)s slov' (14/140 - 15)
 The plural formula is shown as $(n=1) ? 0 : (n>2 \ \&\& \ n<=4) ? 1 : 2$.
- Comments Panel:** Allows users to add comments to the string. It includes a 'New comment' section with a text area and a 'Save' button. The scope is set to 'Translation comment, discussions with other translators'.
- String Information Panel:** Provides details about the string, including its source location ('weblate/templates/translation.html:149'), string age (6 seconds ago), and translation file ('weblate/locale/cs/LC_MESSAGES/django.po, string 5').

Alternative Übersetzungen

Neu in Version 4.13.

Bemerkung: Dies wird derzeit nur mit *Mehrwertige CSV-Datei (UTF-8)* unterstützt.

Bei einigen Formaten ist es möglich, mehrere Übersetzungen für eine einzige Zeichenkette zu verwenden. Sie können weitere alternative Übersetzungen über das Menü *Werkzeuge* hinzufügen. Alle leeren alternativen Übersetzungen werden beim Speichern automatisch entfernt.

Tastaturkürzel

Geändert in Version 2.18: Die Tastaturkürzel wurden in Version 2.18 überarbeitet, um Kollisionen mit Browser- oder Systemvorgaben zu vermeiden.

Die folgenden Tastaturkürzel können während der Übersetzung verwendet werden:

Tastaturkürzel	Beschreibung
Alt+Pos1	Zur ersten Übersetzung in der aktuellen Suche navigieren.
Alt+Ende	Zur letzten Übersetzung in der aktuellen Suche navigieren.
Alt+Bild↑ oder Strg+↑ oder Alt+↑ oder Cmd+↑	Zur vorherigen Übersetzung in der aktuellen Suche navigieren.
Alt+Bild↓ oder Strg+↓ oder Alt+↓ oder Cmd+↓	Zur nächsten Übersetzung in der aktuellen Suche navigieren.
Strg+Eingabe oder Cmd+Eingabe	Aktuelles Formular abschicken; dies entspricht dem Drücken von <i>Speichern und fortfahren</i> während der Bearbeitung der Übersetzung.
Strg+Umschalt+Eingabe oder Cmd+Umschalt+Eingabe	Bearbeitungsmarkierung der Übersetzung aufheben und absenden.
Alt+Eingabe oder Option+Eingabe	Zeichenkette als Übersetzungsvorschlag einreichen; dies entspricht dem Drücken von <i>Vorschlägen</i> während der Bearbeitung der Übersetzung.
Strg+E oder Cmd+E	Übersetzungseditor fokussieren.
Strg+U oder Cmd+U	Kommentareditor fokussieren.
Strg+M oder Cmd+M	Reiterkarte <i>Automatische Vorschläge</i> anzeigen, siehe <i>Automatische Vorschläge</i> .
Strg+1 bis Strg+9 oder Cmd+1 bis Cmd+9	Platzhalter der angegebenen Zahl aus der Ausgangszeichenkette kopieren.
Strg+M+1 bis 9 oder Cmd+M+1 bis 9	Die maschinelle Übersetzung der angegebenen Zahl in die aktuelle Übersetzung kopieren.
Strg+I+1 bis 9 oder Cmd+I+1 bis 9	Ein Element in der Liste der fehlgeschlagenen Qualitätsprüfungen ignorieren.
Strg+J oder Cmd+J	Reiterkarte <i>Benachbarte Zeichenketten</i> anzeigen.
Strg+S oder Cmd+S	Suchfeld fokussieren.
Strg+O oder Cmd+O	Ausgangszeichenkette kopieren.
Strg+Y oder Cmd+Y	Kontrollkästchen <i>Bearbeitung erforderlich</i> umschalten.

Visuelle Tastatur

Eine kleine visuelle Tastaturzeile wird direkt über dem Übersetzungsfeld angezeigt. Dies kann nützlich sein, um die lokale Zeichensetzung im Auge zu behalten (da die Zeile für jede Sprache lokal ist) oder um schwer zu tippende Zeichen zur Hand zu haben.

Die abgebildeten Symbole lassen sich in drei Kategorien einteilen:

- Vom Benutzer konfigurierte *Sonderzeichen*, definiert im *Benutzerprofil*
- Von Weblate bereitgestellte sprachabhängige Zeichen (z. B. Anführungszeichen oder spezifische Zeichen für Sprachen mit linksläufiger Schreibrichtung)
- Mit `SPECIAL_CHARS` konfigurierte Zeichen

The screenshot displays the Weblate web interface for translating a string. The top navigation bar shows 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main area is titled 'WeblateOrg / Django / Hebrew / Translate' and shows a progress bar for 'translated 92%'. The translation area includes a 'Translation' section with 'English' and 'Files' tabs, and a 'Hebrew' tab with a visual keyboard. The keyboard contains various symbols and characters, including 'NBS', 'ZWNJ', 'ZWJ', 'LRM', 'RLM', 'LRE', 'RLE', 'PDF', 'LRO', and 'RLO'. Below the keyboard are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. The right sidebar contains several panels: 'Glossary' (showing 'No related strings found in the glossary'), 'String information' (showing 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'), and 'Nearby strings' (showing a list of strings in Czech, Hungarian, and English).

Übersetzungskontext

Diese kontextbezogene Beschreibung liefert Informationen über die aktuelle Zeichenkette.

Zeichenkette-Attribute

Dinge wie Nachrichten-ID, Kontext (`msgctxt`) oder Position im Quellcode.

Bildschirmfotos

Bildschirmfotos können auf Weblate hochgeladen werden, um Übersetzer besser darüber zu informieren, wo und wie die Zeichenkette verwendet wird, siehe [Visueller Kontext für Zeichenketten](#).

Benachbarte Zeichenketten

Zeigt benachbarte Zeichenketten aus der Übersetzungsdatei an. Diese werden normalerweise auch in einem ähnlichen Kontext verwendet und sind nützlich, um die Übersetzung konsistent zu halten.

Andere Vorkommnisse

Falls eine Zeichenkette an mehreren Stellen erscheint (z. B. bei mehreren Komponenten), zeigt dieser Reiter alle als inkonsistent befundenen an (siehe [Inkonsistent](#)). Sie können auswählen, welche Sie verwenden möchten.

Übersetzungsspeicher

Sehen Sie sich ähnliche Zeichenketten an, die in der Vergangenheit übersetzt wurden, siehe [Übersetzungsspeicher](#).

Glossar

Zeigt die in der aktuellen Zeichenkette verwendeten Begriffe aus dem Projektglossar an.

Letzte Änderungen

Liste der Personen, die diese Zeichenkette kürzlich mit Weblate geändert haben.

Projekt

Project info like instructions for translators, or a directory or link to the string in the version control system repository the project uses.

If you want direct links, the translation format has to support it.

Übersetzungsverlauf

Every change is by default (unless turned off in component settings) saved in the database, and can be reverted. Optionally one can still also revert anything in the underlying version control system.

Translated string length

Weblate can limit the length of a translation in several ways to ensure the translated string is not too long:

- The default limitation for translation is ten times longer than the source string. This can be turned off by [LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH](#). In case you are hitting this, it might be also caused by a monolingual translation erroneously set up as bilingual one, making Weblate mistaking the translation key for the actual source string. See [Zweisprachige und einsprachige Formate](#) for more info.
- Maximal length in characters defined by translation file or flag, see [Maximale Länge der Übersetzung](#).
- Maximal rendered size in pixels defined by flags, see [Maximaler Umfang der Übersetzung](#).

1.3.8 Automatische Vorschläge

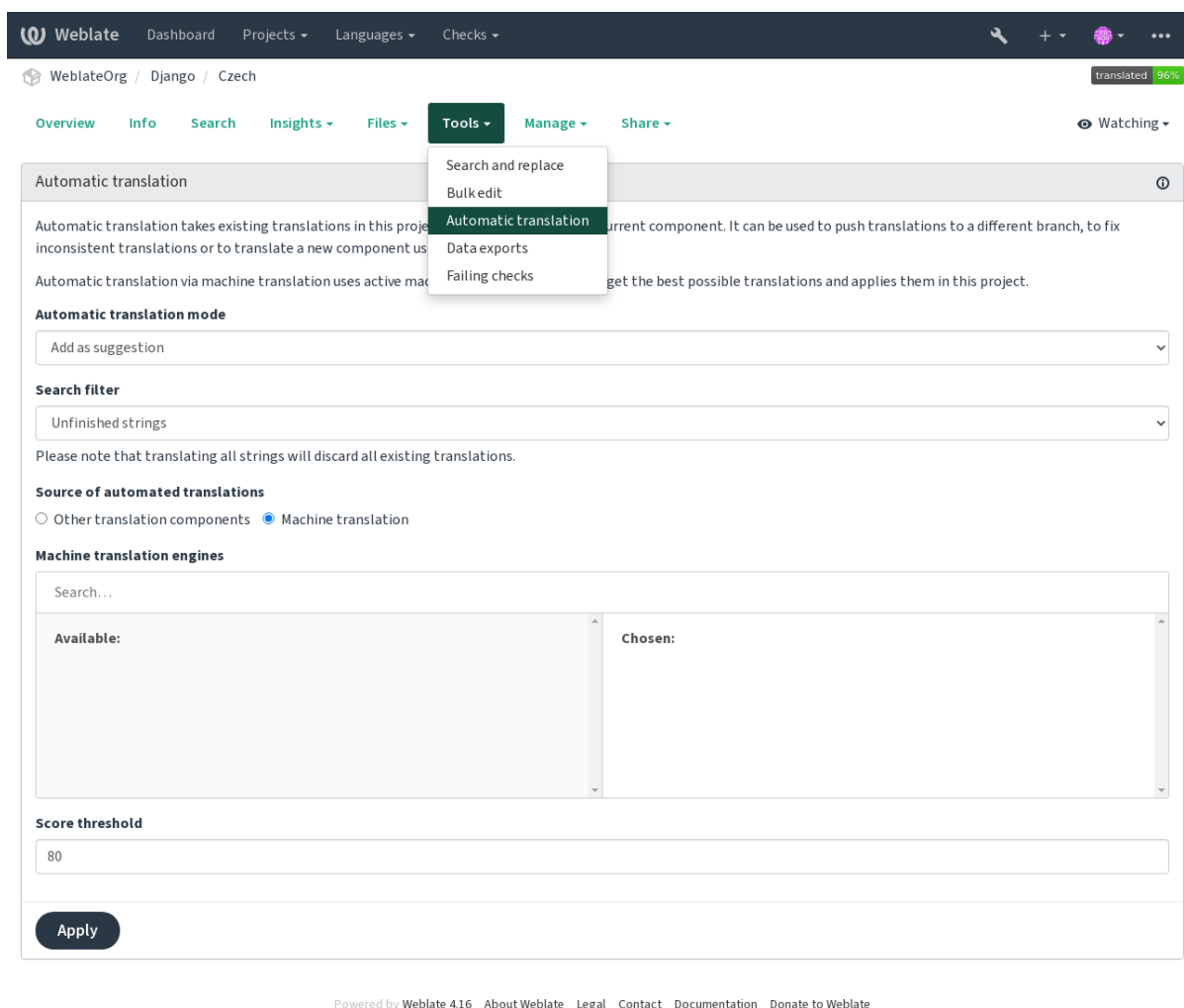
Based on configuration and your translated language, Weblate provides suggestions from several machine translation tools and *Übersetzungsspeicher*. All machine translations are available in a single tab of each translation page.

Siehe auch:

You can find the list of supported tools in *Automatische Vorschläge konfigurieren*.

1.3.9 Automatische Übersetzung

You can use automatic translation to bootstrap translation based on external sources. This tool is called *Automatic translation* accessible in the *Tools* menu, once you have selected a component and a language:



Two modes of operation are possible:

- Using other Weblate components as a source for translations.
- Nutzung ausgewählter maschineller Übersetzungsdienste mit Übersetzungen oberhalb einer bestimmten Qualitätsschwelle.

You can also choose which strings are to be auto-translated.

Warnung: Be mindful that this will overwrite existing translations if employed with wide filters such as *All strings*.

Useful in several situations like consolidating translation between different components (for example the application and its website) or when bootstrapping a translation for a new component using existing translations (translation memory).

Die automatisch übersetzten Zeichenketten sind mit *Automatisch übersetzt* gekennzeichnet.

Siehe auch:

Übersetzungen aller Komponenten gleich halten

1.3.10 Ratenbegrenzung

To avoid abuse of the interface, rate limiting is applied to several operations like searching, sending contact forms or translating. If affected by it, you are blocked for a certain period until you can perform the operation again.

Default limits and fine-tuning is described in the administrative manual, see *Ratenbegrenzung*.

1.3.11 Suchen und Ersetzen

Change terminology effectively or perform bulk fixing of the strings using *Search and replace* in the *Tools* menu.

Hinweis: Machen Sie sich keine Sorgen, dass Sie die Zeichenketten durcheinander bringen. Dies ist ein zweistufiger Prozess, der eine Vorschau der bearbeiteten Zeichenketten anzeigt, bevor die tatsächliche Änderung bestätigt wird.

1.3.12 Massenbearbeitung

Bulk editing allows performing one operation on number of strings. You define strings by searching for them and set up something to be done for matching ones. The following operations are supported:

- Changing string state (for example to approve all unreviewed strings).
- Adjust translation flags (see *Anpassen des Verhaltens mit Markierungen*)
- Adjust string labels (see labels)

Hinweis: This tool is called *Bulk edit* accessible in the *Tools* menu of each project, component or translation.

Siehe auch:

Bulk edit add-on

1.3.13 Matrixansicht

To compare different languages efficiently you can use the matrix view. It is available on every component page under the *Tools* menu. First select all languages you want to compare and confirm your selection, after that you can click on any translation to open and edit it quickly.

The matrix view is also a very good starting point to find missing translations in different languages and quickly add them from one view.

1.3.14 Zen-Modus

The Zen editor can be enabled by clicking the *Zen* button on the top right while translating a component. It simplifies the layout and removes additional UI elements such as *Nearby strings* or the *Glossary*.

You can select the Zen editor as your default editor using the *Einstellungen* tab on your *Benutzerprofil*. Here you can also choose between having translations listed *Top to bottom* or *Side by side* depending on your personal preference.

1.4 Herunterladen und Hochladen von Übersetzungen

Sie können Dateien aus einer Übersetzung exportieren, Änderungen vornehmen und sie dann wieder importieren. So können Sie offline arbeiten und die Änderungen wieder in die bestehende Übersetzung einfügen. Dies funktioniert auch, wenn die Übersetzung in der Zwischenzeit geändert wurde.

Bemerkung: Die verfügbaren Optionen können durch die Einstellungen von der *Zugriffssteuerung* eingeschränkt sein.

1.4.1 Übersetzungen herunterladen

Von der Projekt- oder Komponenten-Übersichtsseite können übersetzbare Dateien über das Menü *Dateien* heruntergeladen werden.

The first option is to download the file in the original format as it is stored in the repository. In this case, any pending changes in the translation are getting committed and the up-to-date file is yield without any conversions.

You can also download the translation converted into one of the widely used localization formats. The converted files will be enriched with data provided in Weblate; such as additional context, comments or flags. Several file formats are available via the *Files* ↓ *Customize download* menu:

- gettext PO
- XLIFF mit Gettext-Erweiterungen
- XLIFF 1.1
- TermBase eXchange
- Translation Memory eXchange
- gettext MO (nur verfügbar, wenn die Übersetzung mit gettext PO erfolgt)
- CSV
- Excel Open XML
- JSON (nur für einsprachige Übersetzungen verfügbar)
- Android String Resource (nur für einsprachige Übersetzungen verfügbar)
- iOS-Zeichenketten (nur für einsprachige Übersetzungen verfügbar)

Hinweis: Der in den konvertierten Dateien verfügbare Inhalt unterscheidet sich aufgrund von Dateiformatmerkmalen, eine Übersicht finden Sie in *Fähigkeiten der Übersetzungstypen*.

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The 'Files' menu is open, showing options: 'Download translation', 'Customize download' (highlighted), and 'Upload translation'. Below the menu is a table of 'Quick downloads' with three rows. The first row is 'File in original format as translated in the repository'. The second row is 'All strings, converted files enriched with comments; suitable for offline translation'. The third row is 'Unfinished strings, converted files enriched with comments; suitable for offline translation'. Each row has columns for 'CSV', 'gettext MO', 'gettext PO', 'TBX', 'TMX', 'XLIFF 1.1 with gettext extensions', 'XLIFF 1.1', and 'XLSX'. Below the table is a 'Customize download' dialog. It has a dropdown for 'All strings' and a 'File format' section with radio buttons for 'gettext PO' (selected), 'XLIFF 1.1 with gettext extensions', 'XLIFF 1.1', 'TBX', 'TMX', 'gettext MO', 'CSV', and 'XLSX'. A 'Download' button is at the bottom of the dialog.

Quick downloads

	CSV	gettext MO	gettext PO	TBX	TMX	XLIFF 1.1 with gettext extensions	XLIFF 1.1	XLSX
26 File in original format as translated in the repository								
26 All strings, converted files enriched with comments; suitable for offline translation								
1 Unfinished strings, converted files enriched with comments; suitable for offline translation								

Customize download

All strings

File format

☒ gettext PO ☐ XLIFF 1.1 with gettext extensions ☐ XLIFF 1.1 ☐ TBX ☐ TMX ☐ gettext MO ☐ CSV ☐ XLSX

Download

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Siehe auch:

`GET /api/translations/(string:project)/(string:component)/(string:language)/file/`

1.4.2 Übersetzungen hochladen

When you have made your changes, use *Upload translation* in the *Files* menu.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb shows 'WeblateOrg / Django / Czech' and a 'translated 96%' indicator. A secondary navigation bar includes 'Overview', 'Info', 'Search', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. The 'Files' dropdown is open, showing 'Download translation', 'Customize download', and 'Upload translation'. The 'Upload' form is displayed below, with a 'File' upload section, 'File upload mode' (radio buttons), 'Processing of "Needs editing" strings' (dropdown), 'Conflict handling' (dropdown), 'Author name' (text field), and 'Author e-mail' (text field). An 'Upload' button is at the bottom of the form. The footer contains 'Powered by Weblate 4.16' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Unterstützte Dateiformate

Any file in a supported file format can be uploaded, but it is still recommended to use the same file format as the one used for translation, otherwise some features might not be translated properly.

Siehe auch:

Unterstützte Dateiformate, Herunterladen und Hochladen von Übersetzungen

Importmethoden

These are the choices presented when uploading translation files:

Als Übersetzung hinzufügen (**translate**)

Imported strings are added as translations to existing strings. This is the most common usecase, and the default behavior.

Es werden nur Übersetzungen aus der hochgeladenen Datei verwendet und keine zusätzlichen Inhalte.

Als Vorschlag hinzufügen (**suggest**)

Imported strings are added as suggestions, do this when you want to have your uploaded strings reviewed.

Es werden nur Übersetzungen aus der hochgeladenen Datei verwendet und keine zusätzlichen Inhalte.

Als fragwürdige Übersetzung hinzufügen (**fuzzy**)

Imported strings are added as translations needing edit. This can be useful when you want translations to be used, but also reviewed.

Es werden nur Übersetzungen aus der hochgeladenen Datei verwendet und keine zusätzlichen Inhalte.

Vorhandene Übersetzungsdatei ersetzen (replace)

Existing file is replaced with new content. This can lead to loss of existing translations, use with caution.

Ausgangszeichenketten aktualisieren (source)

Updates source strings in bilingual translation file. This is similar to what *PO-Dateien auf POT aktualisieren (msgmerge)* does.

Diese Option wird nur für einige Dateiformate unterstützt.

Neue Zeichenketten hinzufügen (add)

Adds new strings to the translation. It skips the one which already exist.

Wenn Sie sowohl neue Zeichenketten hinzufügen als auch bestehende Übersetzungen aktualisieren möchten, laden Sie die Datei ein zweites Mal mit *Als Übersetzung hinzufügen* hoch.

This option is available only with *Zeichenketten verwalten* turned on.

Es werden nur Ausgangszeichenkette, Übersetzung und Schlüssel (Kontext) aus der hochgeladenen Datei verwendet.

Siehe auch:

```
POST /api/translations/(string:project)/(string:component)/(string:language)/file/
```

Konfliktbehandlung

Defines how to deal with uploaded strings which are already translated.

Strings needing edit

There is also an option for how to handle strings needing edit in the imported file. Such strings can be handle in one of the three following ways: „Do not import“, „Import as string needing edit“, or „Import as translated“.

Overriding authorship

With admin permissions, you can also specify authorship of uploaded file. This can be useful in case you've received the file in another way and want to merge it into existing translations while properly crediting the actual author.

1.5 Glossar

Jedes Projekt kann ein oder mehrere Glossare enthalten, in denen die Terminologie gespeichert wird. Das Glossar erleichtert es, die Konsistenz der Übersetzung zu wahren.

Ein Glossar für jede Sprache kann separat verwaltet werden, aber sie werden zusammen als eine einzige Komponente gespeichert, was Projektadministratoren und mehrsprachigen Übersetzern hilft, eine gewisse sprachübergreifende Konsistenz zu wahren. Begriffe aus dem Glossar, die Wörter aus dem aktuell übersetzten Text enthalten, werden in der Seitenleiste des Übersetzungseditors angezeigt.

1.5.1 Verwaltung von Glossaren

Geändert in Version 4.5: Glossare sind jetzt reguläre Übersetzungskomponenten, für die Sie alle Weblate-Funktionen nutzen können – Kommentieren, Speichern in einem entfernten Repository oder Hinzufügen von Erläuterungen.

Verwenden Sie jede Komponente als Glossar, indem Sie *Verwendung als Glossar* aktivieren. Sie können mehrere Glossare für ein Projekt erstellen.

Mit dem Projekt wird automatisch ein leeres Glossar erstellt. Glossare werden von allen Komponenten desselben Projekts gemeinsam genutzt, optional auch von anderen Projekten unter Verwendung von *In Projekten teilen* der jeweiligen Glossarkomponente.

Die Glossarkomponente sieht aus wie jede andere Komponente in Weblate, nur mit einem zusätzlichen farbigen Label:

Translation status

2 Strings 100%

3 Words 100%

Add new glossary term Browse Translate

Strings status

2 All strings — 3 words Browse Translate Zen

2 Translated strings — 3 words Browse Translate Zen

Other components

Component	Translated	Unfinished	Unfinished words	Checks	Suggestions	Comments
Django	96%	1	12	3		
Language names	✓					

Browse all components

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Sie können alle Begriffe des Glossars durchsuchen:

Search interface showing results for 'All strings'.

English	Czech
machine translation	strojový překlad
project	projekt

Add new glossary term

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

oder sie als beliebige Übersetzungen bearbeiten.

1.5.2 Glossarbegriffe

Glossarbegriffe werden auf die gleiche Weise übersetzt wie normale Zeichenketten. Sie können zusätzliche Funktionen über das Menü *Werkzeuge* für jeden Begriff einschalten.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb trail shows 'WeblateOrg / Glossary / WeblateOrg / Czech / Translate'. A status bar indicates 'translated 100%'. The main area is titled 'Glossary term' and contains fields for 'English' (with the value 'project') and 'Czech' (with the value 'projekt'). There are buttons for 'Save and continue', 'Save and stay', 'Suggest', 'Skip', and 'Tools'. A 'Tools' dropdown menu is open, showing options: 'Delete string', 'Mark as untranslatable', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. To the right, a sidebar shows 'Glossary' with a list of terms and 'String information' with details like 'String age' and 'Source string age'. At the bottom, a table shows 'Nearby strings' with columns for 'English' and 'Czech', listing 'machine translation' and 'project'.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Nicht übersetzbare Begriffe

Neu in Version 4.5.

Das Markieren bestimmter Übersetzungen von Glossarbegriffen als *Nur Lesen* durch Massенbearbeitung, Eingabe der Markierung oder durch Verwendung von *Werkzeuge* ↓ *Als nicht übersetzbar markieren* bedeutet, dass sie nicht übersetzt werden können. Verwenden Sie dies für Markennamen oder andere Begriffe, die in anderen Sprachen nicht geändert werden sollten. Solche Begriffe werden in der Seitenleiste des Glossars visuell hervorgehoben.

Siehe auch:

Anpassen des Verhaltens mit Markierungen

Unzulässige Übersetzungen

Neu in Version 4.5.

Das Markieren bestimmter Übersetzungen von Glossarbegriffen als *Unzulässige Übersetzung* durch Massенbearbeitung, Eingabe der Markierung oder durch Verwendung von *Werkzeuge* ↓ *Als unzulässige Übersetzung markieren* bedeutet, dass sie **nicht** verwendet werden dürfen. Verwenden Sie dies, um die Übersetzung klarzustellen, wenn einige Wörter zweideutig sind oder unerwartete Bedeutungen haben könnten.

Siehe auch:

Anpassen des Verhaltens mit Markierungen

Terminologie

Neu in Version 4.5.

Das Markieren bestimmter Glossarbegriffe als *Terminologie* durch Massенbearbeitung, Eingabe der Markierung oder durch Verwendung von *Werkzeuge* ↓ *Als Terminologie markieren* fügt Einträge für diese Begriffe in allen Sprachen des Glossars hinzu. Verwenden Sie dies für wichtige Begriffe, die gut durchdacht sein sollten und in allen Sprachen eine einheitliche Bedeutung haben.

Siehe auch:

Anpassen des Verhaltens mit Markierungen

Varianten

Varianten sind eine allgemeine Möglichkeit, Zeichenketten zusammenzufassen. Alle Begriffsvarianten werden beim Übersetzen in der Seitenleiste des Glossars aufgeführt.

Hinweis: Damit können Sie Abkürzungen oder kürzere Ausdrücke für einen Begriff hinzufügen.

Siehe auch:

variants

1.6 Überprüfungen und Korrekturen

The quality checks help catch common translator errors, ensuring the translation is in good shape. The checks can be ignored in case of false positives.

Once submitting a translation with a failing check, this is immediately shown to the user:

[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)

[Django](#)
[Czech](#)
[Translate](#)

translated 96%

The translation has been saved, however there are some newly failing checks: Missing plurals, Python format

1 / 1

Custom search

'%(count)s word'

Position

1

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

Czech, Few

několik slov

Czech, Many

%(count)s slov

Plural formula:

(n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2

Needs editing

Save and continue

Save and stay

Suggest

Skip

Nearby strings

20

Comments

Automatic suggestions

Other languages

3

History

New comment

Comment on this string for fellow translators and developers to read.

Scope

Translation comment, discussions with other translators

Is your comment specific to this translation, or generic for all of them?

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format

1

Following format strings are missing:

%(count)s

Dismiss

For all languages

Missing plurals

2

Some plural forms are untranslated

Dismiss

For all languages

Glossary

English

Czech

No related strings found in the glossary.

Add term to glossary

String information

1

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

10 seconds ago

Source string age

10 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 5

pending

Powered by Weblate 4.16

[About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

1.6. Überprüfungen und Korrekturen

27

1.6.1 Automatische Korrekturen

In addition to *Qualitätsprüfungen*, Weblate can fix some common errors in translated strings automatically. Use it with caution to not have it add errors.

Siehe auch:

AUTOFIX_LIST

1.6.2 Qualitätsprüfungen

Weblate wendet eine Vielzahl von Qualitätsprüfungen für Zeichenketten an. Der folgende Abschnitt beschreibt sie alle im Detail. Es gibt auch sprachspezifische Prüfungen. Bitte melden Sie einen Fehler, wenn etwas falsch gemeldet wird.

Siehe auch:

CHECK_LIST, *Anpassen des Verhaltens mit Markierungen*

1.6.3 Übersetzungsprüfungen

Wird bei jeder Übersetzungsänderung ausgeführt und hilft den Übersetzern, die Qualität der Übersetzungen aufrechtzuerhalten.

BBCode-Markup

Zusammenfassung

BBCode in der Übersetzung passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.BBCodeCheck`

Prüfungskennung

`bbcode`

Zu ignorierende Markierung

`ignore-bbcode`

BBCode represents simple markup, like for example highlighting important parts of a message in bold font, or italics.

This check ensures they are also found in translation.

Bemerkung: The method for detecting BBCode is currently quite simple so this check might produce false positives.

Aufeinanderfolgende doppelte Wörter

Neu in Version 4.1.

Zusammenfassung

Text enthält zweimal hintereinander dasselbe Wort:

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.duplicate.DuplicateCheck`

Prüfungskennung

duplicate

Zu ignorierende Markierung

ignore-duplicate

Checks that no consecutive duplicate words occur in a translation. This usually indicates a mistake in the translation.

Hinweis: This check includes language specific rules to avoid false positives. In case it triggers falsely in your case, let us know. See [Probleme in Weblate melden](#).

Folgt nicht dem Glossar

Neu in Version 4.5.

Zusammenfassung

Die Übersetzung folgt nicht den definierten Begriffen im Glossar.

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

```
weblate.checks.glossary.GlossaryCheck
```

Prüfungskennung

check_glossary

Flag to enable

check-glossary

Zu ignorierende Markierung

ignore-check-glossary

This check has to be turned on using `check-glossary` flag (see [Anpassen des Verhaltens mit Markierungen](#)). Please consider following prior to enabling it:

- It does exact string matching, the glossary is expected to contain terms in all variants.
- Checking each string against glossary is expensive, it will slow down any operation in Weblate which involves running checks like importing strings or translating.

Siehe auch:

[Glossar](#), [Anpassen des Verhaltens mit Markierungen](#), [Übersetzungsmarkierungen](#)

Doppeltes Leerzeichen

Zusammenfassung

Übersetzung enthält doppelte Leerzeichen

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

```
weblate.checks.chars.DoubleSpaceCheck
```

Prüfungskennung

double_space

Zu ignorierende Markierung

ignore-double-space

Checks that double space is present in translation to avoid false positives on other space-related checks.

Check is false when double space is found in source meaning double space is intentional.

Formatierte Zeichenketten

Checks that formatting in strings are replicated between both source and translation. Omitting format strings in translation usually causes severe problems, so the formatting in strings should usually match the source.

Weblate unterstützt die Überprüfung von Formatzeichenketten in mehreren Sprachen. Die Prüfung wird nicht automatisch aktiviert, sondern nur, wenn eine Zeichenkette entsprechend gekennzeichnet ist (z. B. *c-format* für C-Format). Gettext fügt dies automatisch hinzu, aber Sie müssen es wahrscheinlich manuell für andere Dateiformate hinzufügen oder wenn Ihre PO-Dateien nicht von **xgettext** erzeugt werden.

This can be done per unit (see *Zusätzliche Informationen über Ausgangszeichenketten*) or in *Komponentenkonfiguration*. Having it defined per component is simpler, but can lead to false positives in case the string is not interpreted as a formatting string, but format string syntax happens to be used.

Hinweis: In case specific format check is not available in Weblate, you can use generic *Platzhalter*.

Besides checking, this will also highlight the formatting strings to easily insert them into translated strings:

The screenshot displays the Weblate web interface for managing translations. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The main area shows the translation process for a string in English and Czech. The English string is "%(count)s word" (Singular) and "%(count)s words" (Plural). The Czech translation is currently empty, and the interface shows options to save, suggest, or skip the translation. The sidebar on the right contains a Glossary and String information section.

Translation

English

Singular
%(count)s word

Plural
%(count)s words

Czech, One
%(count)s slovo

Czech, Few
%(count)s slova

Czech, Many
%(count)s slov

Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2

☐ Needs editing

Save and continue **Save and stay** **Suggest** **Skip**

String information

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
No explanation currently provided.

Labels
No labels currently set.

Flags
python-format

Source string location
weblate/templates/translation.html:149

String age
6 seconds ago

Source string age
6 seconds ago

Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5

None
String updated in the repository
WeblateOrg / Django — Czech

English

Singular
%(count)s word

Plural
%(count)s words

Czech 48 characters edited **Current translation** **Translated**

One
%(count)s slovo

Few
%(count)s slova

Many
%(count)s slov

6 seconds ago

[Browse all string changes](#)

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

AngularJS-Interpolations-Zeichenkette

Zusammenfassung

AngularJS-Interpolations-Format-Zeichenketten passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.angularjs.AngularJSInterpolationCheck`

Prüfungskennung

`angularjs_format`

Flag to enable

`angularjs-format`

Zu ignorierende Markierung

`ignore-angularjs-format`

Beispiel für eine benannte Formatzeichenkette

Your balance is {{amount}} {{ currency }}

Siehe auch:

Formatierte Zeichenketten, *AngularJS Text-Interpolation*

C-Format

Zusammenfassung

C-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.CFormatCheck`

Prüfungskennung

`c_format`

Flag to enable

`c-format`

Zu ignorierende Markierung

`ignore-c-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Beispiel für eine Zeichenkette im Positionsformat

Your balance is %1\$d %2\$s

Siehe auch:

Formatierte Zeichenketten,
C-Format-Zeichenketten, C-printf-Format

C#-Format

Zusammenfassung

C#-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.CSharpFormatCheck`

Prüfungskennung

`c_sharp_format`

Flag to enable

`c-sharp-format`

Zu ignorierende Markierung

`ignore-c-sharp-format`

Beispiel für eine Zeichenkette im Positionsformat

There are {0} apples

Siehe auch:*Formatierte Zeichenketten*, *C# String Format***ECMAScript-Buchstabenvorlagen****Zusammenfassung**

ECMAScript-Buchstabenvorlagen passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen`weblate.checks.format.ESTemplateLiteralsCheck`**Prüfungskennung**`es_format`**Flag to enable**`es-format`**Zu ignorierende Markierung**`ignore-es-format`**Beispiel für Interpolation**There are `${number}` apples**Siehe auch:***Formatierte Zeichenketten*, *Template literals***i18next Interpolation**

Neu in Version 4.0.

Zusammenfassung

Die i18next Interpolation passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen`weblate.checks.format.I18NextInterpolationCheck`**Prüfungskennung**`i18next_interpolation`**Flag to enable**`i18next-interpolation`**Zu ignorierende Markierung**`ignore-i18next-interpolation`**Beispiel für Interpolation**There are `{{number}}` apples**Beispiel für Verschachtelung**There are `$t(number)` apples**Siehe auch:***Formatierte Zeichenketten*, *i18next-Interpolation*

ICU MessageFormat

Neu in Version 4.9.

Zusammenfassung

Syntaxfehler und/oder nicht übereinstimmende Platzhalter in ICU MessageFormat-Zeichenketten.

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.icu.ICUMessageFormatCheck`

Prüfungskennung

`icu_message_format`

Flag to enable

`icu-message-format`

Zu ignorierende Markierung

`ignore-icu-message-format`

Beispiel für Interpolation

`There {number, plural, one {is one apple} other {are # apples}}.`

This check has support for both pure ICU MessageFormat messages as well as ICU with simple XML tags. You can configure the behavior of this check by using `icu-flags:*`, either by opting into XML support or by disabling certain sub-checks. For example, the following flag enables XML support while disabling validation of plural sub-messages:

```
icu-message-format, icu-flags:xml:-plural_selectors
```

<code>xml</code>	Enable support for simple XML tags. By default, XML tags are parsed loosely. Stray < characters are ignored if they are not reasonably part of a tag.
<code>strict-xml</code>	Enable support for strict XML tags. All < characters must be escaped if they are not part of a tag.
<code>-highlight</code>	Die Hervorhebung von Platzhaltern im Editor deaktivieren.
<code>-require_other</code>	Disable requiring sub-messages to have an <code>other</code> selector.
<code>-submessage_se</code>	Skip checking that sub-message selectors match the source.
<code>-types</code>	Die Prüfung überspringen, ob Platzhaltertypen mit der Ausgangszeichenkette übereinstimmen.
<code>-extra</code>	Die Prüfung überspringen, ob Platzhalter vorhanden sind, die in der Ausgangszeichenkette nicht vorhanden waren.
<code>-missing</code>	Die Prüfung überspringen, ob keine Platzhalter fehlen, die in der Ausgangszeichenkette vorhanden waren.

Additionally, when `strict-xml` is not enabled but `xml` is enabled, you can use the `icu-tag-prefix:PREFIX` flag to require that all XML tags start with a specific string. For example, the following flag will only allow XML tags to be matched if they start with `<x::`:

```
icu-message-format, icu-flags:xml, icu-tag-prefix:"x:"
```

This would match `<x:link>click here</x:link>` but not `this`.

Siehe auch:

ICU MessageFormat-Syntax, Formatierte Zeichenketten, ICU: Formatting Messages, Format.JS: Message Syntax

Java-Format

Zusammenfassung

Java-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.JavaFormatCheck`

Prüfungskennung

`java_printf_format`

Flag to enable

`java-printf-format`

Zu ignorierende Markierung

`ignore-java-printf-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Beispiel für eine Zeichenkette im Positionsformat

Your balance is %1\$d %2\$s

Geändert in Version 4.14: This used to be toggled by `java-format` flag, it was changed for consistency with GNU gettext.

Siehe auch:

Formatierte Zeichenketten, [Java-Format-Zeichenketten](#)

Java-MessageFormat

Zusammenfassung

Java-MessageFormat-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.JavaMessageFormatCheck`

Prüfungskennung

`java_format`

Markierung zur uncodierten Aktivierung

`java-format`

Markierung zur Aktivierung der automatischen Erkennung

`auto-java-messageformat` aktiviert die Prüfung nur, wenn eine Formatzeichenkette in der Ausgangszeichenkette vorhanden ist

Zu ignorierende Markierung

`ignore-java-format`

Beispiel für eine Zeichenkette im Positionsformat

There are {0} apples

Geändert in Version 4.14: This used to be toggled by `java-messageformat` flag, it was changed for consistency with GNU gettext.

Mit dieser Prüfung wird sichergestellt, dass die Formatzeichenkette für die Java-MessageFormat-Klasse gültig ist. Neben der Überprüfung von Formatzeichenketten in geschweiften Klammern werden auch einfache Anführungszeichen überprüft, da sie eine besondere Bedeutung haben. Wenn ein einfaches Anführungszeichen geschrieben wird,

sollte es als `' '` geschrieben werden. Wenn es nicht gepaart ist, wird es als Beginn des Anführungszeichens behandelt und beim Rendern der Zeichenkette nicht angezeigt.

Siehe auch:

Formatierte Zeichenketten, *Java-MessageFormat*

JavaScript-Format

Zusammenfassung

JavaScript-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.JavaScriptFormatCheck`

Prüfungskennung

`javascript_format`

Flag to enable

`javascript-format`

Zu ignorierende Markierung

`ignore-javascript-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Siehe auch:

Formatierte Zeichenketten, *JavaScript-Format-Zeichenketten*

Lua-Format

Zusammenfassung

Lua-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.LuaFormatCheck`

Prüfungskennung

`lua_format`

Flag to enable

`lua-format`

Zu ignorierende Markierung

`ignore-lua-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Siehe auch:

Formatierte Zeichenketten, *Lua-Format-Zeichenketten*

Object Pascal-Format

Zusammenfassung

Object-Pascal-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.ObjectPascalFormatCheck`

Prüfungskennung

`object_pascal_format`

Flag to enable

`object-pascal-format`

Zu ignorierende Markierung

`ignore-object-pascal-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Siehe auch:

Formatierte Zeichenketten, *Object-Pascal-Format-Zeichenketten*, *Free-Pascal-Format-Zeichenketten* *Delphi-Format-Zeichenketten*

Platzhalter in Prozent

Neu in Version 4.0.

Zusammenfassung

Die prozentualen Platzhalter passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.PercentPlaceholdersCheck`

Prüfungskennung

`percent_placeholders`

Flag to enable

`percent-placeholders`

Zu ignorierende Markierung

`ignore-percent-placeholders`

Beispiel für eine einfache Formatzeichenkette

There are %number% apples

Siehe auch:

Formatierte Zeichenketten,

Perl-Format

Zusammenfassung

Perl-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.PperlFormatCheck`

Prüfungskennung

`perl_format`

Flag to enable

`perl-format`

Zu ignorierende Markierung

`ignore-perl-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Beispiel für eine Zeichenkette im Positionsformat

Your balance is %1\$d %2\$s

Siehe auch:

[Formatierte Zeichenketten](#), [Perl sprintf](#), [Perl-Format-Zeichenketten](#)

PHP-Format

Zusammenfassung

PHP-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.PHPFormatCheck`

Prüfungskennung

`php_format`

Flag to enable

`php-format`

Zu ignorierende Markierung

`ignore-php-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Beispiel für eine Zeichenkette im Positionsformat

Your balance is %1\$d %2\$s

Siehe auch:

[Formatierte Zeichenketten](#), [PHP-sprintf-Dokumentation](#), [PHP-Format-Zeichenketten](#)

Python-Brace-Format

Zusammenfassung

Python-Format-Zeichenkette (geschweifte Klammern) passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.PythonBraceFormatCheck`

Prüfungskennung

`python_brace_format`

Flag to enable

`python-brace-format`

Zu ignorierende Markierung

`ignore-python-brace-format`

Einfache Formatzeichenkette

There are {} apples

Beispiel für eine benannte Formatzeichenkette

Your balance is {amount} {currency}

Siehe auch:

Formatierte Zeichenketten, *Python-Brace-Format*, *Python-Format-Zeichenketten*

Python-Format

Zusammenfassung

Python-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.PythonFormatCheck`

Prüfungskennung

`python_format`

Flag to enable

`python-format`

Zu ignorierende Markierung

`ignore-python-format`

Einfache Formatzeichenkette

There are %d apples

Beispiel für eine benannte Formatzeichenkette

Ihr Saldo beträgt %(amount)d %(currency)s

Siehe auch:

Formatierte Zeichenketten, *Python-String-Formatierung*, *Python-Format-Zeichenketten*

Qt-Format

Zusammenfassung

Qt-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.qt.QtFormatCheck`

Prüfungskennung

`qt_format`

Flag to enable

`qt-format`

Zu ignorierende Markierung

`ignore-qt-format`

Beispiel für eine Zeichenkette im Positionsformat

There are %1 apples

Siehe auch:

Formatierte Zeichenketten, [Qt QString::arg\(\)](#)

Qt-Plural-Format

Zusammenfassung

Qt-Plural-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.qt.QtPluralCheck`

Prüfungskennung

`qt_plural_format`

Flag to enable

`qt-plural-format`

Zu ignorierende Markierung

`ignore-qt-plural-format`

Beispiel für eine Zeichenkette im Pluralformat

There are %Ln apple(s)

Siehe auch:

Formatierte Zeichenketten, [Qt i18n guide](#)

Ruby-Format

Zusammenfassung

Ruby-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.ruby.RubyFormatCheck`

Prüfungskennung

`ruby_format`

Flag to enable

`ruby-format`

Zu ignorierende Markierung

`ignore-ruby-format`

Beispiel für eine einfache Formatzeichenkette

There are %d apples

Beispiel für eine Zeichenkette im Positionsformat

Your balance is %1\$f %2\$s

Beispiel für eine benannte Formatzeichenkette

Your balance is %+.2<amount>f %<currency>s

Named template string

Your balance is %{amount} %{currency}

Siehe auch:

Formatierte Zeichenketten, *Ruby Kernel#sprintf*

Scheme-Format

Zusammenfassung

Scheme-Format-Zeichenkette passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.SchemeFormatCheck`

Prüfungskennung

`scheme_format`

Flag to enable

`scheme-format`

Zu ignorierende Markierung

`ignore-scheme-format`

Beispiel für eine einfache Formatzeichenkette

There are ~d apples

Siehe auch:

Formatierte Zeichenketten, *Srfi 28*, *Chicken Scheme format*, *Guile Scheme formatted output*

Vue I18n-Formatierung

Zusammenfassung

Die Vue I18n-Formatierung passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.format.VueFormattingCheck`

Prüfungskennung

`vue_format`

Flag to enable

`vue-format`

Zu ignorierende Markierung

`ignore-vue-format`

Benannte Formatierung

There are {count} apples

Rails i18n-Formatierung

There are %{count} apples

Verknüpfte lokale Nachrichten

`@:message.dio @:message.the_world!`

Siehe auch:

[Formatierte Zeichenketten](#), [Vue-I18n-Formatierung](#), [Vue I18n Verknüpfte lokale Nachrichten](#)

Ist übersetzt worden

Zusammenfassung

Diese Zeichenkette wurde in der Vergangenheit übersetzt

Zielgruppe

Alle Zeichenketten

Klasse prüfen

`weblate.checks.consistency.TranslatedCheck`

Prüfungskennung

`translated`

Zu ignorierende Markierung

`ignore-translated`

Means a string has been translated already. This can happen when the translations have been reverted in VCS or lost otherwise.

Inkonsistent

Zusammenfassung

Diese Zeichenkette hat in diesem Projekt mehr als eine Übersetzung oder ist in einigen Komponenten nicht übersetzt.

Zielgruppe

Alle Zeichenketten

Klasse prüfen

`weblate.checks.consistency.ConsistencyCheck`

Prüfungskennung

inconsistent

Zu ignorierende Markierung

ignore-inconsistent

Weblate prüft Übersetzungen derselben Zeichenkette für alle Übersetzungen innerhalb eines Projekts, um Ihnen zu helfen, konsistente Übersetzungen zu erhalten.

The check fails on differing translations of one string within a project. This can also lead to inconsistencies in displayed checks. You can find other translations of this string on the *Other occurrences* tab.

This check applies to all components in a project that have *Verbreitung von Übersetzungen erlauben* turned on.

Hinweis: Aus Leistungsgründen findet die Prüfung möglicherweise nicht alle Inkonsistenzen, so dass die Anzahl der Übereinstimmungen begrenzt ist.

Bemerkung: This check also fires in case the string is translated in one component and not in another. It can be used as a quick way to manually handle strings which are untranslated in some components just by clicking on the *Use this translation* button displayed on each line in the *Other occurrences* tab.

You can use *Automatische Übersetzung* add-on to automate translating of newly added strings which are already translated in another component.

Siehe auch:

Übersetzungen aller Komponenten gleich halten

Kashida-Buchstabe verwendet

Neu in Version 3.5.

Zusammenfassung

Dekorative Kashida-Schriftzüge sollten nicht verwendet werden

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.KashidaCheck`

Prüfungskennung

kashida

Zu ignorierende Markierung

ignore-kashida

Die dekorativen Kashida-Buchstaben sollten in der Übersetzung nicht verwendet werden. Diese sind auch als Tatweel bekannt.

Siehe auch:

[Kashida auf Wikipedia](#)

Markdown-Links

Neu in Version 3.5.

Zusammenfassung

Markdown-Links in der Übersetzung passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.MarkdownLinkCheck`

Prüfungskennung

`md-link`

Flag to enable

`md-text`

Zu ignorierende Markierung

`ignore-md-link`

Markdown links do not match source.

Siehe auch:

[Markdown links](#)

Markdown-Referenzen

Neu in Version 3.5.

Zusammenfassung

Markdown-Link-Referenzen passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.MarkdownRefLinkCheck`

Prüfungskennung

`md-reflink`

Flag to enable

`md-text`

Zu ignorierende Markierung

`ignore-md-reflink`

Markdown link references do not match source.

Siehe auch:

[Markdown links](#)

Markdown-Syntax

Neu in Version 3.5.

Zusammenfassung

Markdown-Syntax passt nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.MarkdownSyntaxCheck`

Prüfungskennung

`md-syntax`

Flag to enable

`md-text`

Zu ignorierende Markierung

`ignore-md-syntax`

Markdown-Syntax passt nicht zur Ausgangszeichenkette

Siehe auch:

[Markdown span elements](#)

Maximale Länge der Übersetzung

Zusammenfassung

Die Übersetzung sollte die angegebene Länge nicht überschreiten

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.MaxLengthCheck`

Prüfungskennung

`max-length`

Flag to enable

`max-length`

Zu ignorierende Markierung

`ignore-max-length`

Checks that translations are of acceptable length to fit available space. This only checks for the length of translation characters.

Unlike the other checks, the flag should be set as a `key:value` pair like `max-length:100`.

Hinweis: This check looks at number of chars, what might not be the best metric when using proportional fonts to render the text. The *Maximaler Umfang der Übersetzung* check does check actual rendering of the text.

The `replacements:` flag might be also useful to expand placeables before checking the string.

When `xml-text` flag is also used, the length calculation ignores XML tags.

Maximaler Umfang der Übersetzung

Zusammenfassung

Der übersetzte, gerenderte Text sollte die vorgegebene Größe nicht überschreiten

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.render.MaxSizeCheck`

Prüfungskennung

`max-size`

Flag to enable

`max-size`

Zu ignorierende Markierung

`ignore-max-size`

Neu in Version 3.7.

Translation rendered text should not exceed given size. It renders the text with line wrapping and checks if it fits into given boundaries.

This check needs one or two parameters - maximal width and maximal number of lines. In case the number of lines is not provided, one line text is considered.

You can also configure used font by `font-*` directives (see [Anpassen des Verhaltens mit Markierungen](#)), for example following translation flags say that the text rendered with ubuntu font size 22 should fit into two lines and 500 pixels:

```
max-size:500:2, font-family:ubuntu, font-size:22
```

Hinweis: You might want to set `font-*` directives in [Komponentenkonfiguration](#) to have the same font configured for all strings within a component. You can override those values per string in case you need to customize it per string.

The `replacements:` flag might be also useful to expand placeables before checking the string.

When `xml-text` flag is also used, the length calculation ignores XML tags.

Siehe auch:

[Schriftarten verwalten](#), [Anpassen des Verhaltens mit Markierungen](#), [Maximale Länge der Übersetzung](#)

Unterschiedliche Menge \n

Zusammenfassung

Anzahl der `\n`-Symbole in der Übersetzung passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EscapedNewlineCountingCheck`

Prüfungskennung

`escaped_newline`

Zu ignorierende Markierung

`ignore-escaped-newline`

Normalerweise sind maskierte Zeilenumbrüche wichtig für die Formatierung der Programmausgabe. Die Prüfung schlägt fehl, wenn die Anzahl der `\n`-Symbole in der Übersetzung nicht zur Ausgangszeichenkette passt.

Nicht übereinstimmender Doppelpunkt

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette enden nicht beide mit einem Doppelpunkt

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndColonCheck`

Prüfungskennung

`end_colon`

Zu ignorierende Markierung

`ignore-end-colon`

Checks that colons are replicated between both source and translation. The presence of colons is also checked for various languages where they do not belong (Chinese or Japanese).

Siehe auch:

[Doppelpunkt auf Wikipedia](#)

Nicht übereinstimmende Auslassungspunkte

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette enden nicht beide mit Auslassungspunkten (...)

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndEllipsisCheck`

Prüfungskennung

`end_ellipsis`

Zu ignorierende Markierung

`ignore-end-ellipsis`

Prüft, ob nachgestellte Auslassungspunkte in der Ausgangszeichenkette und in der Übersetzung wiedergegeben werden. Dies gilt nur für echte Auslassungspunkte (...), nicht für drei Punkte (. . .).

Auslassungspunkte werden in der Regel schöner als drei Punkte in der Druckschrift dargestellt und hören sich bei Text-to-Speech besser an.

Siehe auch:

[Auslassungspunkte auf Wikipedia](#)

Nicht übereinstimmendes Ausrufezeichen

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette enden nicht beide mit einem Ausrufezeichen

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndExclamationCheck`

Prüfungskennung

`end_exclamation`

Zu ignorierende Markierung

`ignore-end-exclamation`

Checks that exclamations are replicated between both source and translation. The presence of exclamation marks is also checked for various languages where they do not belong (Chinese, Japanese, Korean, Armenian, Limbu, Myanmar or Nko).

Siehe auch:

[Ausrufezeichen auf Wikipedia](#)

Nicht übereinstimmender Punkt

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette enden nicht beide mit einem Punkt

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndStopCheck`

Prüfungskennung

`end_stop`

Zu ignorierende Markierung

`ignore-end-stop`

Checks that full stops are replicated between both source and translation. The presence of full stops is checked for various languages where they do not belong (Chinese, Japanese, Devanagari or Urdu).

Siehe auch:

[Punkt auf Wikipedia](#)

Nicht übereinstimmendes Fragezeichen

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette enden nicht beide mit einem Fragezeichen

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndQuestionCheck`

Prüfungskennung

`end_question`

Zu ignorierende Markierung

`ignore-end-question`

Checks that question marks are replicated between both source and translation. The presence of question marks is also checked for various languages where they do not belong (Armenian, Arabic, Chinese, Korean, Japanese, Ethiopic, Vai or Coptic).

Siehe auch:

[Fragezeichen auf Wikipedia](#)

Nicht übereinstimmendes Semikolon

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette enden nicht beide mit einem Semikolon

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndSemicolonCheck`

Prüfungskennung

`end_semicolon`

Zu ignorierende Markierung

`ignore-end-semicolon`

Überprüft, ob Semikolons am Ende von Sätzen sowohl in der Ausgangszeichenkette als auch in der Übersetzung wiedergegeben werden.

Siehe auch:

[Semikolon auf Wikipedia](#)

Nicht übereinstimmende Zeilenumbrüche

Zusammenfassung

Anzahl der neuen Zeilen in der Übersetzung passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.NewLineCountCheck`

Prüfungskennung

`newline-count`

Zu ignorierende Markierung

`ignore-newline-count`

Normalerweise sind Zeilenumbrüche wichtig für die Formatierung der Programmausgabe. Die Prüfung schlägt fehl, wenn die Anzahl der `\n`-Symbole in der Übersetzung nicht zur Ausgangszeichenkette passt.

Fehlende Pluralformen

Zusammenfassung

Einige Pluralformen sind nicht übersetzt

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.consistency.PluralsCheck`

Prüfungskennung

`plurals`

Zu ignorierende Markierung

`ignore-plurals`

Checks that all plural forms of a source string have been translated. Specifics on how each plural form is used can be found in the string definition.

Failing to fill in plural forms will in some cases lead to displaying nothing when the plural form is in use.

Platzhalter

Neu in Version 3.9.

Zusammenfassung

Der Übersetzung fehlen einige Platzhalter

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.placeholders.PlaceholderCheck`

Prüfungskennung

`placeholders`

Flag to enable

`placeholders`

Zu ignorierende Markierung

`ignore-placeholders`

Geändert in Version 4.3: Sie können einen regulären Ausdruck als Platzhalter verwenden.

Geändert in Version 4.13: With the `case-insensitive` flag, the placeholders are not case-sensitive.

Translation is missing some placeholders. These are either extracted from the translation file or defined manually using `placeholders` flag, more can be separated with colon, strings with space can be quoted:

```
placeholders:$URL$:$TARGET$:"some long text"
```

In case you have some syntax for placeholders, you can use a regular expression:

```
placeholders:r"%[^\% ]%"
```

You can also have case insensitive placeholders:

```
placeholders:$URL$:$TARGET$,case-insensitive
```

Siehe auch:

Anpassen des Verhaltens mit Markierungen

Satzzeichenabstand

Neu in Version 3.9.

Zusammenfassung

Es fehlt ein untrennbares Leerzeichen vor doppelten Satzzeichen

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.PunctuationSpacingCheck`

Prüfungskennung

`punctuation_spacing`

Zu ignorierende Markierung

`ignore-punctuation-spacing`

Prüft, ob vor doppelten Interpunktionszeichen (Ausrufezeichen, Fragezeichen, Semikolon und Doppelpunkt) ein nicht abbrechbares Leerzeichen steht. Diese Regel wird nur in einigen ausgewählten Sprachen wie Französisch oder Bretonisch verwendet, wo das Leerzeichen vor doppelten Satzzeichen eine typografische Regel ist.

Siehe auch:

[Französische und englische Abstände auf Wikipedia](#)

Regulärer Ausdruck

Neu in Version 3.9.

Zusammenfassung

Übersetzung stimmt nicht mit regulärem Ausdruck überein

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.placeholders.RegexCheck`

Prüfungskennung

`regex`

Flag to enable

`regex`

Zu ignorierende Markierung

`ignore-regex`

Translation does not match regular expression. The expression is either extracted from the translation file or defined manually using `regex` flag:

```
regex: ^foo|bar$
```

Identische Pluralformen**Zusammenfassung**

Einige Pluralformen sind identisch übersetzt

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.consistency.SamePluralsCheck`

Prüfungskennung

`same-plurals`

Zu ignorierende Markierung

`ignore-same-plurals`

Check that fails if some plural forms are duplicated in the translation. In most languages they have to be different.

Zeilenumbruch am Anfang**Zusammenfassung**

Ausgangs- und übersetzte Zeichenkette beginnen nicht beide mit einem Zeilenumbruch

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.BeginNewlineCheck`

Prüfungskennung

`begin_newline`

Zu ignorierende Markierung

`ignore-begin-newline`

Zeilenumbrüche erscheinen normalerweise aus gutem Grund in den Ausgangszeichenketten, denn fehlende oder hinzugefügte Zeilenumbrüche können zu Formatierungsproblemen führen, wenn der übersetzte Text verwendet wird.

Siehe auch:

[Zeilenumbruch am Ende](#)

Leerzeichen am Anfang

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette beginnen mit unterschiedlich vielen Leerzeichen

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.BeginSpaceCheck`

Prüfungskennung

`begin_space`

Zu ignorierende Markierung

`ignore-begin-space`

Ein Leerzeichen am Anfang einer Zeichenkette wird üblicherweise für Einrückungen in der Darstellung verwendet und sollte deshalb beibehalten werden.

Zeilenumbruch am Ende

Zusammenfassung

Quelle und Übersetzung enden nicht beide mit einem Zeilenumbruch

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndNewlineCheck`

Prüfungskennung

`end_newline`

Zu ignorierende Markierung

`ignore-end-newline`

Zeilenumbrüche erscheinen normalerweise aus gutem Grund in den Ausgangszeichenketten, denn fehlende oder hinzugefügte Zeilenumbrüche können zu Formatierungsproblemen führen, wenn der übersetzte Text verwendet wird.

Siehe auch:

[Zeilenumbruch am Anfang](#)

Leerzeichen am Ende

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette enden nicht beide mit einem Leerzeichen

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.EndSpaceCheck`

Prüfungskennung

`end_space`

Zu ignorierende Markierung

`ignore-end-space`

Überprüft, ob die Leerzeichen am Ende des Textes zwischen Ausgangszeichenkette und Übersetzung wiedergegeben werden.

Leerzeichen am Ende werden in der Regel verwendet, um benachbarte Elemente voneinander abzugrenzen, so dass ihre Entfernung das Layout beeinträchtigen könnte.

Unveränderte Übersetzung

Zusammenfassung

Ausgangs- und übersetzte Zeichenkette sind identisch

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.same.SameCheck`

Prüfungskennung

`same`

Zu ignorierende Markierung

`ignore-same`

Tritt ein, wenn der Ausgangstext und die entsprechende Übersetzung identisch sind, bis hin zu mindestens einer der Pluralformen. Einige Zeichenketten, die in allen Sprachen vorkommen, werden ignoriert, und verschiedene Auszeichnungselemente werden entfernt. Dadurch wird die Anzahl der falsch-positiven Ergebnisse reduziert.

This check can help find strings mistakenly untranslated.

Das Standardverhalten dieser Prüfung ist, dass Wörter aus der eingebauten schwarzen Liste von der Prüfung ausgeschlossen werden. Dies sind Wörter, die häufig nicht übersetzt werden. Dies ist nützlich, um Fehlalarme bei kurzen Zeichenketten zu vermeiden, die nur aus einem einzigen Wort bestehen, das in mehreren Sprachen gleich ist. Diese schwarze Liste kann durch Hinzufügen der Markierung `strict-same` zur Zeichenkette oder Komponente deaktiviert werden.

Siehe auch:

Komponentenkonfiguration, Anpassen des Verhaltens mit Markierungen

Unsicheres HTML

Neu in Version 3.9.

Zusammenfassung

Die Übersetzung verwendet unsichere HTML-Markierungen

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.SafeHTMLCheck`

Prüfungskennung

`safe-html`

Flag to enable

`safe-html`

Zu ignorierende Markierung

`ignore-safe-html`

The translation uses unsafe HTML markup. This check has to be enabled using `safe-html` flag (see [Anpassen des Verhaltens mit Markierungen](#)). There is also accompanied autofixer which can automatically sanitize the markup.

Hinweis: When `md-text` flag is also used, the Markdown style links are also allowed.

Siehe auch:

The HTML check is performed by the [Ammonia](#) library.

URL

Neu in Version 3.5.

Zusammenfassung

Die Übersetzung enthält keine URL

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.URLCheck`

Prüfungskennung

`url`

Flag to enable

`url`

Zu ignorierende Markierung

`ignore-url`

Die Übersetzung enthält keine URL. Dies wird nur ausgelöst, wenn die Einheit als URL-haltig markiert ist. In diesem Fall muss die Übersetzung eine gültige URL sein.

XML-Auszeichnung

Zusammenfassung

XML-Tags in der Übersetzung passen nicht zur Ausgangszeichenkette

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.XMLTagsCheck`

Prüfungskennung

`xml-tags`

Zu ignorierende Markierung

`ignore-xml-tags`

Das bedeutet normalerweise, dass die Ausgabe anders aussieht. In den meisten Fällen ist dies kein erwünschtes Ergebnis einer Änderung der Übersetzung, aber gelegentlich schon.

Checks that XML tags are replicated between both source and translation.

Bemerkung: Diese Prüfung wird durch die Markierung `safe-html` deaktiviert, da die von ihr durchgeführte HTML-Bereinigung HTML-Markup erzeugen kann, das kein gültiges XML ist.

XML-Syntax

Neu in Version 2.8.

Zusammenfassung

Die Übersetzung ist kein gültiges XML

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.markup.XMLValidityCheck`

Prüfungskennung

`xml-invalid`

Zu ignorierende Markierung

`ignore-xml-invalid`

The XML markup is not valid.

Bemerkung: Diese Prüfung wird durch die Markierung `safe-html` deaktiviert, da die von ihr durchgeführte HTML-Bereinigung HTML-Markup erzeugen kann, das kein gültiges XML ist.

Leerzeichen ohne Breite

Zusammenfassung

Übersetzung enthält zusätzliche Leerzeichen ohne Breite

Zielgruppe

Übersetzte Zeichenketten

Klasse prüfen

`weblate.checks.chars.ZeroWidthSpaceCheck`

Prüfungskennung

`zero-width-space`

Zu ignorierende Markierung

`ignore-zero-width-space`

Zero-width space (<U+200B>) characters are used to break messages within words (word wrapping).

As they are usually inserted by mistake, this check is triggered once they are present in translation. Some programs might have problems when this character is used.

Siehe auch:

[Leerzeichen ohne Breite auf Wikipedia](#)

1.6.4 Source checks

Source checks can help developers improve the quality of source strings.

Auslassungspunkte

Zusammenfassung

Die Zeichenkette verwendet drei Punkte (...) anstelle der Auslassungspunkte (...)

Zielgruppe

Ausgangszeichenketten

Klasse prüfen

`weblate.checks.source.EllipsisCheck`

Prüfungskennung

`ellipsis`

Zu ignorierende Markierung

`ignore-ellipsis`

Dies schlägt fehl, wenn die Zeichenkette drei Punkte enthält (. . .), obwohl sie Auslassungspunkte verwenden sollte (...).

Die Verwendung von Unicode-Zeichen ist in den meisten Fällen der bessere Ansatz, sieht besser aus und hört sich mit Text-to-Speech vermutlich besser an.

Siehe auch:

[Auslassungspunkte auf Wikipedia](#)

ICU MessageFormat-Syntax

Neu in Version 4.9.

Zusammenfassung

Syntaxfehler in ICU MessageFormat-Zeichenketten.

Zielgruppe

Ausgangszeichenketten

Klasse prüfen

`weblate.checks.icu.ICUSourceCheck`

Prüfungskennung

`icu_message_format_syntax`

Flag to enable

`icu-message-format`

Zu ignorierende Markierung

`ignore-icu-message-format`

Siehe auch:

ICU MessageFormat

Lange nicht übersetzt

Neu in Version 4.1.

Zusammenfassung

Die Zeichenkette wurde lange Zeit nicht übersetzt

Zielgruppe

Ausgangszeichenketten

Klasse prüfen

`weblate.checks.source.LongUntranslatedCheck`

Prüfungskennung

`long_untranslated`

Zu ignorierende Markierung

`ignore-long-untranslated`

When the string has not been translated for a long time, it can indicate a problem in a source string making it hard to translate.

Mehrfach fehlgeschlagene Überprüfungen

Zusammenfassung

In mehreren Sprachen liegen bei den Übersetzungen fehlgeschlagene Qualitätsprüfungen vor

Zielgruppe

Ausgangszeichenketten

Klasse prüfen

`weblate.checks.source.MultipleFailingCheck`

Prüfungskennung

`multiple_failures`

Zu ignorierende Markierung

`ignore-multiple-failures`

Numerous translations of this string have failing quality checks. This is usually an indication that something could be done to improve the source string.

This check failing can quite often be caused by a missing full stop at the end of a sentence, or similar minor issues which translators tend to fix in translation, while it would be better to fix it in the source string.

Mehrere unbenannte Variablen

Neu in Version 4.1.

Zusammenfassung

Es existieren mehrere unbenannte Variablen in der Zeichenkette, so dass es für Übersetzer unmöglich ist, sie neu zu ordnen

Zielgruppe

Ausgangszeichenketten

Klasse prüfen

`weblate.checks.format.MultipleUnnamedFormatsCheck`

Prüfungskennung

`unnamed_format`

Zu ignorierende Markierung

`ignore-unnamed-format`

There are multiple unnamed variables in the string, making it impossible for translators to reorder them.

Consider using named variables instead to allow translators to reorder them.

Ohne Pluralformen

Zusammenfassung

Die Zeichenkette wird im Plural verwendet, hat aber keine Pluralformen

Zielgruppe

Ausgangszeichenketten

Klasse prüfen

`weblate.checks.source.OptionalPluralCheck`

Prüfungskennung

`optional_plural`

Zu ignorierende Markierung

`ignore-optional-plural`

The string is used as a plural, but does not use plural forms. In case your translation system supports this, you should use the plural aware variant of it.

For example with Gettext in Python it could be:

```
from gettext import ngettext

print(ngettext("Selected %d file", "Selected %d files", files) % files)
```

1.7 Suche

Neu in Version 3.9.

Erweiterte Abfragen mit booleschen Operationen, Klammern oder feldspezifischen Suchabfragen können verwendet werden, um die gewünschten Zeichenketten zu finden.

Wenn kein Feld definiert ist, erfolgt die Suche anhand von Ausgangs-, Ziel- und Kontextzeichenketten.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a 'Dashboard' section with 'Watched translations' (0), 'Suggested translations' (0), 'Insights', and a 'Search' button. The main content area is titled 'Search' and contains a search bar with 'All strings' selected and a 'Sort By' dropdown. Below the search bar is the 'Advanced query builder' section with a 'Source strings' dropdown, a 'Search for...' input, and buttons for 'Exact', 'Add', 'Strings with suggestions', and 'Add'. There's also a 'String changed after' section with a date input 'mm/dd/yyyy' and an 'Add' button. The 'Query examples' section lists several predefined queries with their corresponding filter expressions and 'Add' buttons:

Query examples	Filter expression	Action
Review strings changed by other users	<code>changed:>=2023-01-29 AND NOT changed_by:testuser</code>	Add
Translated strings	<code>state:>=translated</code>	Add
Strings with comments	<code>has:comment</code>	Add
Strings with any failing checks	<code>has:check</code>	Add
Strings with suggestions from others	<code>has:suggestion AND NOT suggestion_author:testuser</code>	Add
Approved strings with suggestions	<code>state:approved AND has:suggestion</code>	Add
All untranslated strings added the past month	<code>added:>=2023-01-29 AND state:<=needs-editing</code>	Add
Translated strings in a certain language	<code>is:translated AND language:cs</code>	Add

At the bottom of the search section is a 'Search' button.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.7.1 Einfache Suche

Jede in das Suchfeld eingegebene Phrase wird in Wörter zerlegt. Zeichenketten, die eines dieser Wörter enthalten, werden angezeigt. Um nach einem exakten Ausdruck zu suchen, setzen Sie „die Suchphrase“ in Anführungszeichen (sowohl einfache (‘) als auch doppelte (“) Anführungszeichen sind möglich): "dies ist eine Zeichenkette in Anführungszeichen" oder 'eine andere Zeichenkette in Anführungszeichen'.

1.7.2 Felder

source:TEXT

Suche in Ausgangszeichenkette ohne Berücksichtigung der Groß-/Kleinschreibung.

target:TEXT

Suche in Zielzeichenkette ohne Berücksichtigung der Groß-/Kleinschreibung.

context:TEXT

Suche in Kontextzeichenkette ohne Berücksichtigung der Groß-/Kleinschreibung.

key:TEXT

Suche in Schlüsselzeichenkette ohne Berücksichtigung der Groß-/Kleinschreibung.

note:TEXT

Suche in Beschreibung der Ausgangszeichenkette ohne Berücksichtigung der Groß-/Kleinschreibung.

location:TEXT

Suche in Standortzeichenkette ohne Berücksichtigung der Groß- und Kleinschreibung.

priority:NUMMER

Priorität der Zeichenkette.

id:NUMBER

String unique identifier.

added:DATUMZEIT

Zeitstempel, wann die Zeichenkette zu Weblate hinzugefügt wurde.

state:TEXT

Suche nach Zeichenkettenstatus (approved, translated, needs-editing, empty, read-only), unterstützt *Feld-Operatoren*.

pending:BOOLESCH

Zeichenkette, die zur Übergabe an das VCS ansteht.

has:TEXT

Suche nach Zeichenkette mit den Attributen – plural, context, suggestion, comment, check, dismissed-check, translation, variant, screenshot, flags, explanation, glossary, note, label.

is:TEXT

Suche nach ausstehenden Übersetzungen (pending). Kann auch nach allen Zeichenkettenstatus suchen (approved, translated, untranslated, needs-editing, read-only).

language:TEXT

Zielsprache der Zeichenkette.

component:TEXT

Suche nach Komponenten-Slug oder -Name ohne Berücksichtigung der Groß-/Kleinschreibung, siehe *Komponenten-Slug* und *Name der Komponente*.

project:TEXT

Projekt-Slug, siehe *URL-Kurzbegriff*.

changed_by:TEXT

Zeichenkette wurde vom Autor mit dem angegebenen Benutzernamen geändert.

changed:DATUMZEIT

Änderungsdatum des Zeichenketteninhalts, unterstützt *Feld-Operatoren*.

change_time:DATUMZEIT

Änderungsdatum der Zeichenkette, unterstützt *Feld-Operatoren*, im Gegensatz zu `changed` umfasst dies Ereignisse, die den Inhalt nicht ändern und Sie können eine benutzerdefinierte Filterung nach Aktionen mit `change_action` anwenden.

change_action:TEXT

Filtert nach Änderungsaktionen, nützlich zusammen mit `change_time`. Akzeptiert den englischen Namen der Änderungsaktion, entweder in Anführungszeichen und mit Leerzeichen oder in Kleinbuchstaben und Leerzeichen die durch einen Bindestrich ersetzt werden. Siehe *Nach Änderungen suchen* für Beispiele.

check:TEXT

Prüfung der Zeichenkette ist fehlgeschlagen, siehe *Überprüfungen und Korrekturen* für Prüfkennungen.

dismissed_check:TEXT

Prüfung der Zeichenkette wurde abgelehnt, siehe *Überprüfungen und Korrekturen* für Prüfkennungen.

comment:TEXT

Suche in Benutzerkommentaren.

resolved_comment:TEXT

Suche in geklärten Kommentaren.

comment_author:TEXT

Nach Autor des Kommentars filtern.

suggestion:TEXT

Suche in Vorschlägen.

suggestion_author:TEXT

Nach Autor des Vorschlags filtern.

explanation:TEXT

Suche in Erklärungen.

label:TEXT

Suche in Labels.

screenshot:TEXT

Suche in Bildschirmfotos.

1.7.3 Boolesche Operatoren

Sie können Suchabfragen mit AND, OR, NOT und Klammern kombinieren, um komplexe Abfragen zu bilden. Zum Beispiel: `state:translated AND (source:hello OR source:bar)`

1.7.4 Feld-Operatoren

Sie können Operatoren, Bereiche oder partielle Suchabfragen für Datums- oder numerische Suchen angeben:

state:>=translated

Status ist `translated` oder besser (`approved`).

changed:2019

Geändert im Jahr 2019.

changed:[2019-03-01 to 2019-04-01]

Geändert zwischen zwei Datumsangaben.

1.7.5 Genaue Operatoren

Mit dem Operator `=` können Sie eine exakte Abfrage für verschiedene Zeichenkettenfelder durchführen. Um zum Beispiel nach allen Ausgangszeichenketten zu suchen, die genau mit `hello world` übereinstimmen, verwenden Sie: `source:="hello world"`. Für die Suche nach einzelnen Wortausdrücken können Sie die Anführungszeichen weglassen. Um zum Beispiel nach allen Ausgangszeichenketten zu suchen, die mit `hello` übereinstimmen, können Sie `source:=hello` verwenden.

1.7.6 Nach Änderungen suchen

Neu in Version 4.4.

Die Suche nach Ereignissen im Verlauf kann mit den Operatoren `change_action` und `change_time` durchgeführt werden.

Die Suche nach Zeichenketten, die im Jahr 2018 zur Bearbeitung markiert wurden, kann beispielsweise als `change_time:2018 AND change_action:marked-for-edit` oder `change_time:2018 AND change_action:"Marked for edit"` eingegeben werden.

1.7.7 Reguläre Ausdrücke

Überall dort, wo Text akzeptiert wird, können Sie auch einen regulären Ausdruck als `r"regexp"` angeben.

Um zum Beispiel nach allen Ausgangszeichenketten zu suchen, die eine beliebige Ziffer zwischen 2 und 5 enthalten, verwenden Sie `source:r"[2-5]"`.

1.7.8 Vordefinierte Abfragen

Auf der Suchseite können Sie aus vordefinierten Suchanfragen auswählen und so schnell auf die häufigsten Suchanfragen zugreifen:

The screenshot displays the Weblate web interface for a Django project. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The sidebar on the left shows the project structure: WeblateOrg / Django / Czech / Translate. The main content area features a search filter 'Custom search' with the value '%(count)s word'. A dropdown menu is open, showing various filters for string states: Untranslated strings (state:empty), Unfinished strings (state:<translated), Translated strings (state:>=translated), Strings marked for edit (state:needs-editing), Strings with suggestions (has:suggestion), Strings with variants (has:variant), Strings with screenshots (has:screenshot), Strings with labels (has:label), Strings with context (has:context), Unfinished strings without suggestions (state:<translated AND NOT has:suggestion), Strings with comments (has:comment), Strings with any failing checks (has:check), Approved strings (state:approved), and Strings waiting for review (state:translated). The right sidebar shows string information: Screenshot context (No screenshot currently associated), Explanation (No explanation currently provided), Labels (No labels currently set), Flags (python-format), Source string location (weblate/templates/translation.html:149), String age (6 seconds ago), Source string age (6 seconds ago), and Translation file (weblate/locale/cs/LC_MESSAGES/django.po, string 5).

1.7.9 Sortieren der Ergebnisse

Es gibt viele Möglichkeiten, die Zeichenketten nach Ihren Bedürfnissen zu sortieren:

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation form for a string: 'The string uses three dots (...) instead of an ellipsis character (...)'. The form includes fields for the source string (English) and the target string (Czech). A dropdown menu is open over the 'Position and priority' button, showing options like Position, Priority, Labels, Source string, Target string, String age, Number of words, Number of comments, Number of failing checks, Key, and String location. The sidebar on the right contains a 'Glossary' section and a 'String information' section. The 'String information' section shows details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'. The bottom section is for 'New comment'.

Powered by Weblate 4.16 · [About Weblate](#) · [Legal](#) · [Contact](#) · [Documentation](#) · [Donate to Weblate](#)

1.8 Übersetzungsabläufe

Der Einsatz von Weblate bedeutet mehr Nähe zwischen Ihnen und Ihren Benutzern und damit auch mehr Nähe zwischen Ihnen und Ihren Übersetzern. Es liegt an Ihnen zu entscheiden, wie viele der Funktionen Sie nutzen möchten.

Die folgende Liste ist keine vollständige Auflistung der Konfigurationsmöglichkeiten von Weblate. Sie können weitere Workflows auf der Grundlage der hier aufgelisteten gängigen Beispiele erstellen.

1.8.1 Zugriff auf die Übersetzung

The *access control* is not discussed in detail as a whole in the workflows, as most of its options can be applied to any workflow. Please consult the respective documentation on how to manage access to translations.

In den folgenden Kapiteln bedeutet *jeder Benutzer* einen Benutzer, der Zugriff auf die Übersetzung hat. Das kann jeder authentifizierte Benutzer sein, wenn das Projekt öffentlich ist, oder ein Benutzer, der eine *Übersetzen*-Berechtigung für das Projekt hat.

1.8.2 Übersetzungsstatus

Jede übersetzte Zeichenkette kann sich in einem der folgenden Zustände befinden:

Nicht übersetzt

Die Übersetzung ist leer, je nach Dateiformat kann sie in der Datei gespeichert werden oder nicht.

Bearbeitung erforderlich

Die Übersetzung muss bearbeitet werden. Dies ist normalerweise das Ergebnis einer Änderung der Ausgangszeichenkette, eines Fuzzy-Matching oder einer Aktion des Übersetzers. Die Übersetzung wird in der Datei gespeichert, je nach Dateiformat kann sie als bearbeitungsbedürftig gekennzeichnet sein (z. B. durch eine fuzzy-Markierung in der Gettext-Datei).

Ausstehende Überprüfungen

Die Übersetzung wird vorgenommen, aber nicht überprüft. Sie wird in der Datei als eine gültige Übersetzung gespeichert.

Genehmigt

Die Übersetzung wurde in der Überprüfung genehmigt. Sie kann nicht mehr von Übersetzern geändert werden, sondern nur noch von Prüfern. Übersetzer können nur noch Vorschläge einreichen.

This state is only available when reviews are enabled.

Vorschläge

Vorschläge werden nur in Weblate und nicht in der Übersetzungsdatei gespeichert.

Die Zustände werden soweit möglich in den Übersetzungsdateien dargestellt.

Hinweis: Falls das von Ihnen verwendete Dateiformat das Speichern von Zuständen nicht unterstützt, sollten Sie die Erweiterung *Unveränderte Übersetzungen als „bearbeitungsbedürftig“ markieren* verwenden, um unveränderte Zeichenketten als zu bearbeiten zu kennzeichnen.

Siehe auch:

Fähigkeiten der Übersetzungstypen, Übersetzungsabläufe

1.8.3 Direkte Übersetzung

Dies ist die übliche Einstellung für kleinere Teams, in denen jeder direkt übersetzen kann. Dies ist auch die Standardeinstellung in Weblate.

- *Jeder Benutzer* kann Übersetzungen bearbeiten.
- Vorschläge sind eine optionale Möglichkeit, Änderungen vorzuschlagen, wenn sich die Übersetzer über die Änderung unsicher sind.

Einstellung	Value	Anmerkung
Überprüfung aktivieren	aus	Auf Projektebene konfiguriert.
Vorschläge aktivieren	an	Es ist nützlich für die Benutzer, Vorschläge machen zu können, wenn sie unsicher sind.
Abstimmen über Vorschläge	aus	
Vorschläge automatisch annehmen	0	
Übersetzergruppe	<i>Benutzer</i>	Oder <i>Übersetzen</i> mit <i>projektspezifischer Zugriffssteuerung</i> .
Prüfergruppe	N/A	Nicht verwendet.

1.8.4 Peer-Review

In diesem Arbeitsablauf kann jeder Übersetzungsvorschläge einbringen, die von weiteren Mitgliedern genehmigt werden müssen, bevor sie als Übersetzung akzeptiert werden.

- *Jeder Benutzer* kann Übersetzungsvorschläge hinzufügen.
- *Jeder Benutzer* kann für Übersetzungsvorschläge abstimmen.
- Vorschläge werden zu Übersetzungen, wenn sie eine bestimmte Anzahl von Stimmen erhalten haben.

Einstellung	Value	Anmerkung
Überprüfung aktivieren	aus	Auf Projektebene konfiguriert.
Vorschläge aktivieren	an	
Abstimmen über Vorschläge	aus	
Vorschläge automatisch annehmen	1	Sie können einen höheren Wert einstellen, um mehr Peer-Reviews anzufordern.
Übersetzergruppe	<i>Benutzer</i>	Oder <i>Übersetzen</i> mit <i>projektspezifischer Zugriffssteuerung</i> .
Prüfergruppe	N/A	Nicht verwendet, alle Übersetzer überprüfen.

1.8.5 Zugehörige Prüfer

Neu in Version 2.18: Der korrekte Überprüfungsablauf wird seit Weblate 2.18 unterstützt.

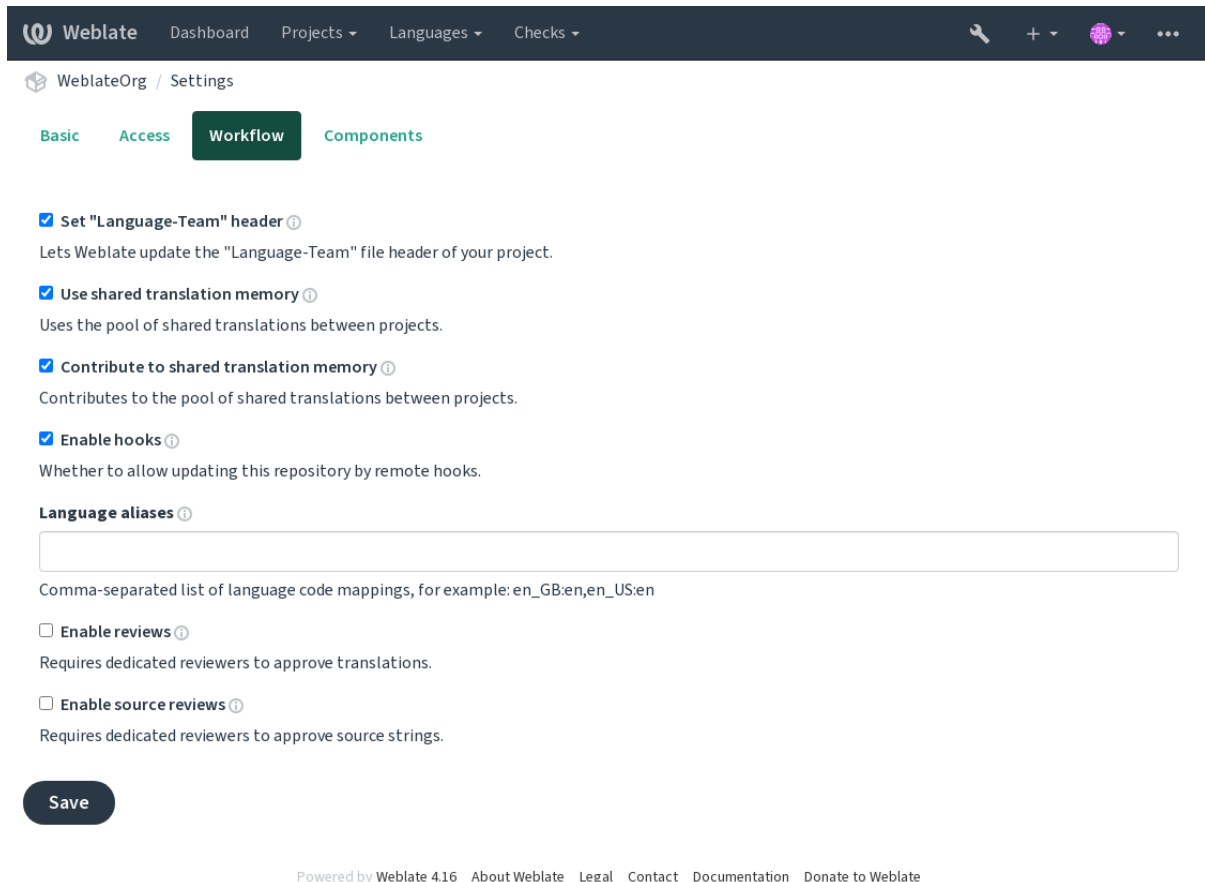
Mit speziellen Prüfern haben Sie zwei Benutzergruppen, von denen eine Übersetzungen einreichen kann und die andere sie überprüft, um sicherzustellen, dass die Übersetzungen konsistent sind und die Qualität stimmt.

- *Jeder Benutzer* kann nicht genehmigte Übersetzungen bearbeiten.
- *Prüfer* kann Zeichenketten genehmigen / nicht genehmigen.
- *Prüfer* können alle Übersetzungen (einschließlich genehmigten) bearbeiten.
- Vorschläge können auch verwendet werden, um Änderungen für genehmigte Zeichenketten vorzuschlagen.

Einstellung	Value	Anmerkung
Überprüfung aktivieren	an	Auf Projektebene konfiguriert.
Vorschläge aktivieren	aus	Es ist nützlich für die Benutzer, Vorschläge machen zu können, wenn sie unsicher sind.
Abstimmen über Vorschläge	aus	
Vorschläge automatisch annehmen	0	
Übersetzergruppe	<i>Benutzer</i>	Oder <i>Übersetzen</i> mit <i>projektspezifischer Zugriffssteuerung</i> .
Prüfergruppe	<i>Prüfer</i>	Oder <i>Überprüfen</i> mit <i>projektspezifischer Zugriffssteuerung</i> .

1.8.6 Einschalten von Überprüfungen

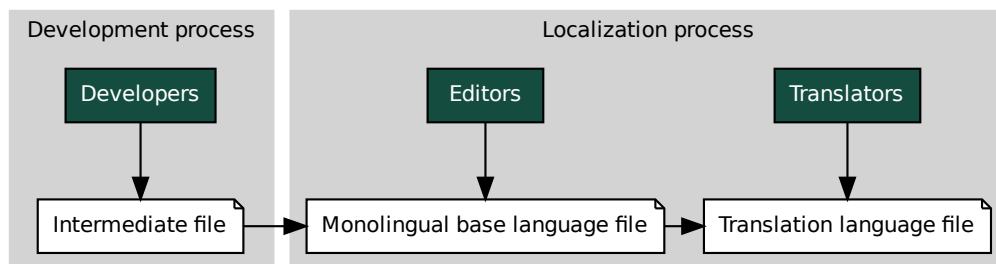
Überprüfungen können in der Projektkonfiguration auf der Unterseite *Arbeitsablauf* der Projekteinstellungen aktiviert werden (zu finden im Menü *Verwalten* → *Einstellungen*):



1.8.7 Qualitäts-Gateway für die Ausgangszeichenketten

In vielen Fällen stammen die ursprünglichen Zeichenketten der Ausgangssprache von den Entwicklern, da sie den Code schreiben und die ursprünglichen Zeichenketten bereitstellen. Die Entwickler sind jedoch oft keine Muttersprachler in der Ausgangssprache und bieten nicht die gewünschte Qualität der Ausgangszeichenketten. Die Zwischenübersetzung kann Ihnen dabei helfen, dies zu beheben – es gibt eine zusätzliche Qualitätskontrolle für die Texte zwischen Entwicklern, Übersetzern und Benutzern.

Wenn Sie *Zwischensprachdatei* einstellen, wird diese Datei als Quelle für die Zeichenketten verwendet, aber sie wird in der Ausgangssprache bearbeitet, um sie zu optimieren. Sobald die Zeichenkette in der Ausgangssprache fertig ist, steht sie auch für Übersetzer zur Verfügung, die sie in weitere Sprachen übersetzen.



Siehe auch:

Zwischensprachdatei, Einsprachige Basis-Sprachdatei, Zweisprachige und einsprachige Formate

1.8.8 Überprüfungen der Ausgangszeichenketten

Wenn *Quellenüberprüfung aktivieren* aktiviert ist, kann der Überprüfungsprozess auf die Ausgangszeichenketten angewendet werden. Sobald er aktiviert ist, können Benutzer Probleme in den Ausgangszeichenketten melden. Der tatsächliche Prozess hängt davon ab, ob Sie ein- oder zweisprachige Formate verwenden.

Bei einsprachigen Formaten verhält sich die Überprüfung der Ausgangszeichenkette ähnlich wie bei *Zugehörige Prüfer* – sobald ein Problem mit der Ausgangszeichenkette gemeldet wird, wird sie als *Bearbeitung erforderlich* markiert.

Die zweisprachigen Formate erlauben keine direkte Bearbeitung der Ausgangszeichenketten (diese werden normalerweise direkt aus dem Quellcode extrahiert). In diesem Fall wird das Label *Ausgangszeichenkette muss überprüft werden* an die Zeichenketten angehängt, die von Übersetzern gemeldet werden. Sie sollten solche Zeichenketten überprüfen und sie entweder im Quellcode bearbeiten oder die Kennzeichnung entfernen.

Siehe auch:

Zweisprachige und einsprachige Formate, Zugehörige Prüfer, labels, Kommentare

1.9 Häufig gestellte Fragen

1.9.1 Konfiguration

Wie erstellt man einen automatisierten Arbeitsablauf?

Weblate can handle all the translation things semi-automatically for you. If you give it push access to your repository, the translations can happen without interaction, unless some merge conflict occurs.

1. Set up your Git repository to tell Weblate when there is any change, see *Benachrichtigungs-Hooks* for info on how to do it.
2. Set a push URL at your *Komponentenkonfiguration* in Weblate, this allows Weblate to push changes to your repository.
3. Turn on *Bei Commit gleichzeitig Pushen* on your *Komponentenkonfiguration* in Weblate, this will make Weblate push changes to your repository whenever they happen at Weblate.

Siehe auch:

Kontinuierliche Lokalisierung, Vermeiden von Merge-Konflikten

Wie greift man über SSH auf Repositorys zu?

Bitte lesen Sie *Zugriff auf Repositorys* für Informationen zur Einrichtung von SSH-Schlüsseln.

Wie behebt man Merge-Konflikte in Übersetzungen?

Merge conflicts happen from time to time when the translation file is changed in both Weblate and the upstream repository concurrently. You can usually avoid this by merging Weblate translations prior to making changes in the translation files (e.g. before running msgmerge). Just tell Weblate to commit all pending translations (you can do it in *Repository maintenance* in the *Manage* menu) and merge the repository (if automatic push is not on).

Wenn Sie bereits auf einen Merge-Konflikt gestoßen sind, ist es am einfachsten, alle Konflikte lokal auf Ihrem Rechner zu lösen, indem Sie Weblate als Remote-Repository hinzufügen, es mit Upstream zusammenführen und alle Konflikte beheben. Sobald Sie die Änderungen pushen, kann Weblate die zusammengeführte Version ohne weitere besondere Maßnahmen verwenden.

Bemerkung: Depending on your setup, access to the Weblate repository might require authentication. When using the built-in *Git-Exporter* in Weblate, you authenticate with your username and the API key.

```
# Commit all pending changes in Weblate, you can do this in the UI as well:
wlc commit
# Lock the translation in Weblate, again this can be done in the UI as well:
wlc lock
# Add Weblate as remote:
git remote add weblate https://hosted.weblate.org/git/project/component/
# You might need to include credentials in some cases:
git remote add weblate https://username:APIKEY@hosted.weblate.org/git/project/
↪component/

# Update weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

# Resolve conflicts:
edit ...
git add ...
...
git commit

# Rebase changes (if Weblate is configured to do rebases)
git rebase origin/main

# Push changes to upstream repository, Weblate will fetch merge from there:
git push

# Open Weblate for translation:
wlc unlock
```

If you're using multiple branches in Weblate, you can do the same to all of them:

```
# Add and update Weblate remotes
git remote add weblate-one https://hosted.weblate.org/git/project/one/
git remote add weblate-second https://hosted.weblate.org/git/project/second/
git remote update weblate-one weblate-second

# Merge QA_4_7 branch:
git checkout QA_4_7
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
git merge weblate-one/QA_4_7
... # Resolve conflicts
git commit

# Merge main branch:
git checkout main
git merge weblates-second/main
... # Resolve conflicts
git commit

# Push changes to the upstream repository, Weblate will fetch the merge from there:
git push
```

In case of gettext PO files, there is a way to merge conflicts in a semi-automatic way:

Fetch and keep a local clone of the Weblate Git repository. Also get a second fresh local clone of the upstream Git repository (i. e. you need two copies of the upstream Git repository: An intact and a working copy):

```
# Add remote:
git remote add weblate /path/to/weblate/snapshot/

# Update Weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

# Resolve conflicts in the PO files:
for PO in `find . -name '*.po'` ; do
    msgcat --use-first /path/to/weblate/snapshot/$PO\
              /path/to/upstream/snapshot/$PO -o $PO.merge
    msgmerge --previous --lang=${PO%.po} $PO.merge domain.pot -o $PO
    rm $PO.merge
    git add $PO
done
git commit

# Push changes to the upstream repository, Weblate will fetch merge from there:
git push
```

Siehe auch:

Wie exportiert man das von Weblate verwendete Git-Repository?, Kontinuierliche Lokalisierung, Vermeiden von Merge-Konflikten, Weblate-Client

Wie übersetzt man mehrere Branches auf einmal?

Weblate unterstützt das Pushen von Übersetzungsänderungen innerhalb eines Projekts. Für jede Komponente, die dies aktiviert hat (Standardeinstellung), wird die vorgenommene Änderung automatisch an andere weitergegeben. Auf diese Weise werden die Übersetzungen synchron gehalten, auch wenn die Zweige selbst schon sehr weit auseinander liegen und es nicht möglich ist, Übersetzungsänderungen einfach zwischen ihnen zusammenzuführen.

Once you merge changes from Weblate, you might have to merge these branches (depending on your development workflow) discarding differences:

```
git merge -s ours origin/maintenance
```

Siehe auch:

Übersetzungen aller Komponenten gleich halten

Wie übersetzt man plattformübergreifende Projekte?

Weblate supports a wide range of file formats (see [Unterstützte Dateiformate](#)) and the easiest approach is to use the native format for each platform.

Once you have added all platform translation files as components in one project (see [Adding translation projects and components](#)), you can utilize the translation propagation feature (turned on by default, and can be turned off in the [Komponentenkonfiguration](#)) to translate strings for all platforms at once.

Siehe auch:

Übersetzungen aller Komponenten gleich halten

Wie exportiert man das von Weblate verwendete Git-Repository?

There is nothing special about the repository, it lives under the `DATA_DIR` directory and is named `vcs/<project>/<component>/`. If you have SSH access to this machine, you can use the repository directly.

For anonymous access, you might want to run a Git server and let it serve the repository to the outside world.

Alternatively, you can use [Git-Exporter](#) inside Weblate to automate this.

Welche Möglichkeiten gibt es, um Änderungen wieder Upstream zu pushen?

This heavily depends on your setup, Weblate is quite flexible in this area. Here are examples of some workflows used with Weblate:

- Weblate automatically pushes and merges changes (see [Wie erstellt man einen automatisierten Arbeitsablauf?](#)).
- You manually tell Weblate to push (it needs push access to the upstream repository).
- Somebody manually merges changes from the Weblate git repository into the upstream repository.
- Somebody rewrites history produced by Weblate (e.g. by eliminating merge commits), merges changes, and tells Weblate to reset the content in the upstream repository.

Of course you are free to mix all of these as you wish.

Wie beschränkt man den Weblate-Zugriff auf Übersetzungen, ohne den Quellcode freizugeben?

You can use [git submodule](#) for separating translations from source code while still having them under version control.

1. Create a repository with your translation files.
2. Add this as a submodule to your code:

```
git submodule add git@example.com:project-translations.git path/to/translations
```

3. Link Weblate to this repository, it no longer needs access to the repository containing your source code.
4. You can update the main repository with translations from Weblate by:

```
git submodule update --remote path/to/translations
```

Please consult the [git submodule](#) documentation for more details.

Wie überprüft man ob Weblate richtig eingerichtet ist?

Weblate enthält eine Reihe von Konfigurationsprüfungen, die Sie in der Adminoberfläche sehen können. Folgen Sie einfach dem Link *Leistungsbericht* in der Adminoberfläche oder öffnen Sie direkt die URL `/manage/performance/`.

Siehe auch:

Weblate überwachen, Celery-Status überwachen

Warum werden alle Commits von Weblate <noreply@weblate.org> übertragen?

This is the default committer name, configured by `DEFAULT_COMMITTER_EMAIL` and `DEFAULT_COMMITTER_NAME`.

The author of every commit (if the underlying VCS supports it) is still recorded correctly as the user that made the translation.

For commits where no authorship is known (for example anonymous suggestions or machine translation results), the authorship is credited to the anonymous user (see `ANONYMOUS_USER_NAME`). You can change the name and e-mail in the management interface.

Siehe auch:

Komponentenkonfiguration

Wie verschiebt man Dateien im Repository, ohne den Verlauf in Weblate zu verlieren?

Um den Verlauf, die Kommentare oder Bildschirmfotos, die mit Zeichenketten verknüpft sind, nach einer Änderung des Dateispeicherorts beizubehalten, müssen Sie sicherstellen, dass diese Zeichenketten in Weblate niemals gelöscht werden. Dies kann passieren, wenn das Weblate-Repository aktualisiert wird, die Komponentenkonfiguration aber noch auf die alten Dateien verweist. Dies führt dazu, dass Weblate davon ausgeht, dass es alle Übersetzungen löschen sollte.

The solution to this is to perform the operation in sync with Weblate:

1. Sperren Sie die betroffene Komponente in Weblate.
2. Comitten Sie alle ausstehenden Änderungen und führen Sie sie in das Upstream-Repository ein.
3. Deaktivieren Sie den Empfang von Webhooks für die *Projektkonfiguration*; dies verhindert, dass Weblate Änderungen im Repository sofort sieht.
4. Führen Sie alle notwendigen Änderungen im Projektarchiv durch (z. B. mit `git mv`) und pushen Sie sie in das Upstream-Repository.
5. Ändern Sie die *Komponentenkonfiguration* so, dass sie mit der neuen Konfiguration übereinstimmt; wenn Sie die Konfiguration ändern, wird Weblate das aktualisierte Repository abrufen und die geänderten Orte bemerken, während die bestehenden Zeichenketten erhalten bleiben.
6. Entsperren Sie die Komponente und aktivieren Sie die Hooks in der Projektkonfiguration wieder.

1.9.2 Anwendung

Wie überprüft man die Übersetzungen anderer?

- Es gibt mehrere überprüfungsbasierte Arbeitsabläufe in Weblate, siehe :ref:`workflows`.
- Sie können alle Änderungen, die in Abonnements gemacht werden, abonnieren und dann andere Beiträge, die per E-Mail eintreffen, überprüfen.
- Am unteren Rand der Übersichtsansicht steht ein Überprüfungsstool zur Verfügung, mit dem Sie Übersetzungen durchsuchen können, die seit einem bestimmten Datum von anderen erstellt wurden.

Siehe auch:

Übersetzungsabläufe

Wie gibt man Rückmeldung zu einer Ausgangszeichenkette?

On context tabs below translation, you can use the *Comments* tab to provide feedback on a source string, or discuss it with other translators.

Siehe auch:

report-source, *Kommentare*

Wie verwendet man vorhandene Übersetzungen beim Übersetzen?

- Dank des gemeinsamen Übersetzungsspeichers können alle Übersetzungen innerhalb von Weblate verwendet werden.
- Sie können vorhandene Übersetzungsspeicherdateien in Weblate importieren.
- Use the import functionality to load compendium as translations, suggestions or translations needing review. This is the best approach for a one-time translation using a compendium or a similar translation database.
- You can set up *tmserver* with all databases you have and let Weblate use it. This is good when you want to use it several times during translation.
- Another option is to translate all related projects in a single Weblate instance, which will make it automatically pick up translations from other projects as well.

Siehe auch:

Automatische Vorschläge konfigurieren, Automatische Vorschläge, Übersetzungsspeicher

Aktualisiert Weblate neben den Übersetzungen auch die Übersetzungsdateien?

Weblate tries to limit changes in translation files to a minimum. For some file formats it might unfortunately lead to reformatting the file. If you want to keep the file formatted your way, please use a pre-commit hook for that.

Siehe auch:

updating-target-files

Woher kommen die Sprachdefinitionen und wie fügt man eigene Definitionen hinzu?

The basic set of language definitions is included within Weblate and Translate-toolkit. This covers more than 150 languages and includes info about plural forms or text direction.

You are free to define your own languages in the administrative interface, you just need to provide info about it.

Siehe auch:

Sprachdefinitionen

Kann Weblate Änderungen in einer fragwürdigen Zeichenkette hervorheben?

Weblate supports this, however it needs the data to show the difference.

For Gettext PO files, you have to pass the parameter `--previous` to **msgmerge** when updating PO files, for example:

```
msgmerge --previous -U po/cs.po po/phpmyadmin.pot
```

For monolingual translations, Weblate can find the previous string by ID, so it shows the differences automatically.

Warum zeigt Weblate immer noch alte Übersetzungszeichenketten an, obwohl die Vorlage aktualisiert wurde?

Weblate does not try to manipulate the translation files in any way other than allowing translators to translate. So it also does not update the translatable files when the template or source code have been changed. You simply have to do this manually and push changes to the repository, Weblate will then pick up the changes automatically.

Bemerkung: It is usually a good idea to merge changes done in Weblate before updating translation files, as otherwise you will usually end up with some conflicts to merge.

For example with gettext PO files, you can update the translation files using the **msgmerge** tool:

```
msgmerge -U locale/cs/LC_MESSAGES/django.mo locale/django.pot
```

In case you want to do the update automatically, you can install add-on *PO-Dateien auf POT aktualisieren (msgmerge)*.

Siehe auch:

updating-target-files

1.9.3 Fehlerbehebung

Anfragen schlagen manchmal mit der Fehlermeldung „zu viele offene Dateien“ fehl

This happens sometimes when your Git repository grows too much and you have many of them. Compressing the Git repositories will improve this situation.

The easiest way to do this is to run:

```
# Go to DATA_DIR directory
cd data/vcs
# Compress all Git repositories
for d in */* ; do
    pushd $d
    git gc
    popd
done
```

Siehe auch:

[DATA_DIR](#)

Beim Zugriff auf die Website erscheint die Fehlermeldung „Fehlerhafte Anfrage (400)“

This is most likely caused by an improperly configured `ALLOWED_HOSTS`. It needs to contain all hostnames you want to access on your Weblate. For example:

```
ALLOWED_HOSTS = ["weblate.example.com", "weblate", "localhost"]
```

Siehe auch:

[Einrichtung zulässiger Hosts](#)

Was bedeutet „Es gibt mehrere Dateien für die einzelne Sprache (en)“?

This typically happens when you have translation file for source language. Weblate keeps track of source strings and reserves source language for this. The additional file for same language is not processed.

- Falls die Übersetzung in die Ausgangssprache gewünscht ist, ändern Sie bitte die [Ausgangssprache](#) in den Komponenteneinstellungen. Vielleicht möchten Sie *Englisch (Entwickler)* als Ausgangssprache verwenden, oder das [Qualitäts-Gateway für die Ausgangszeichenketten](#) nutzen.
- Falls die Übersetzungsdatei für die Ausgangssprache nicht benötigt wird, entfernen Sie diese bitte aus dem Repository.
- Falls die Übersetzungsdatei für die Ausgangssprache benötigt wird, aber von Weblate ignoriert werden soll, stellen Sie bitte den [Sprachen-Filter](#) so ein, dass er sie ausschließt.

Hinweis: You might get similar error message for other languages as well. In that case the most likely reason is that several files map to single language in Weblate.

This can be caused by using obsolete language codes together with new one (`ja` and `jp` for Japanese) or including both country specific and generic codes (`fr` and `fr_FR`). See [Parsing von Sprachcodes](#) for more details.

1.9.4 Funktionen

Unterstützt Weblate auch andere VCS als Git und Mercurial?

Weblate currently does not have native support for anything other than [Git](#) (with extended support for [GitHub-Pull-Requests](#), [Gerrit](#) and [Subversion](#)) and [Mercurial](#), but it is possible to write backends for other VCSes.

You can also use [Git remote helpers](#) in Git to access other VCSes.

Weblate also supports VCS-less operation, see [Lokale Dateien](#).

Bemerkung: For native support of other VCSes, Weblate requires using distributed VCS, and could probably be adjusted to work with anything other than Git and Mercurial, but somebody has to implement this support.

Siehe auch:

[Integration der Versionsverwaltung](#)

Wie erfolgt der Übersetzungsnachweis bei Weblate?

Every change made in Weblate is committed into VCS under the translators name. This way every single change has proper authorship, and you can track it down using the standard VCS tools you use for code.

Additionally, when the translation file format supports it, the file headers are updated to include the translator's name.

Siehe auch:

list_translators, *../devel/reporting*

Warum erzwingt Weblate die Anzeige aller PO-Dateien in einem einzigen Baum?

Weblate was designed in a way that every PO file is represented as a single component. This is beneficial for translators, so they know what they are actually translating.

Geändert in Version 4.2: Übersetzer können alle Komponenten eines Projekts als Ganzes in eine bestimmte Sprache übersetzen.

Warum verwendet Weblate Sprachcodes wie sr_Latn oder zh_Hant?

Dies sind Sprachcodes, die von [RFC 5646](#) definiert wurden, um besser anzuzeigen, dass es sich wirklich um verschiedene Sprachen handelt, anstatt der bisher falsch verwendeten Modifikatoren (für `@latin`-Varianten) oder Ländercodes (für Chinesisch).

Weblate versteht immer noch die herkömmlichen Sprachcodes und wird sie auf die aktuellen abbilden – zum Beispiel wird `sr@latin` als `sr_Latn` oder `zh@CN` als `zh_Hans` behandelt.

Bemerkung: Weblate verwendet standardmäßig Sprachcodes im POSIX-Stil mit Unterstrich, siehe *[Sprachdefinitionen](#)* für weitere Details.

Siehe auch:

Sprachdefinitionen, *Stil des Sprachcodes*, *Neue Übersetzungen hinzufügen*

1.10 Unterstützte Dateiformate

Weblate supports most translation format understood by [translate-toolkit](#), however each format being slightly different, some issues with formats that are not well tested can arise.

Siehe auch:

[Translation Related File Formats](#)

Bemerkung: When choosing a file format for your application, it's better to stick some well established format in the toolkit/platform you use. This way your translators can additionally use whatever tools they are used to, and will more likely contribute to your project.

1.10.1 Zweisprachige und einsprachige Formate

Both monolingual and bilingual formats are supported. Bilingual formats store two languages in single file—source and translation (typical examples are *GNU gettext*, *XLIFF* or *Apple iOS-Zeichenketten*). On the other side, monolingual formats identify the string by ID, and each language file contains only the mapping of those to any given language (typically *Android String-Ressourcen*). Some file formats are used in both variants, see the detailed description below.

For correct use of monolingual files, Weblate requires access to a file containing complete list of strings to translate with their source—this file is called *Einsprachige Basis-Sprachdatei* within Weblate, though the naming might vary in your paradigm.

Additionally this workflow can be extended by utilizing *Zwischensprachdatei* to include strings provided by developers, but not to be used as is in the final strings.

1.10.2 Automatische Erkennung

Weblate can automatically detect several widespread file formats, but this detection can harm your performance and will limit features specific to given file format (for example automatic addition of new translations).

1.10.3 Fähigkeiten der Übersetzungstypen

Fähigkeiten aller unterstützten Formate:

Format	Lingualität ^{Seite 79, 1}	Pluralformen ^{Seite 79, 2}	Beschreibungen ^{Seite 79, 3}	Kontext ^{Seite 79, 4}	Ort ^{Seite 79, 5}	Markierungen ^{Seite 79, 8}	Zusätzliche Status ^{Seite 79, 6}
<i>GNU gettext</i>	zweisprachig	ja	ja	ja	ja	ja ⁹	Bearbeitung erforderlich
<i>Mono-lingual gettext</i>	einsprachig	ja	ja	ja	ja	ja ⁹	Bearbeitung erforderlich
<i>XLIFF</i>	beides	ja	ja	ja	ja	ja ¹⁰	Bearbeitung erforderlich, genehmigt
<i>Java properties</i>	beides	nein	ja	nein	nein	nein	
<i>mi18n lang Dateien</i>	einsprachig	nein	ja	nein	nein	nein	
<i>GWT-Eigenschaften</i>	einsprachig	ja	ja	nein	nein	nein	
<i>Joomla-Übersetzung</i>	einsprachig	nein	ja	nein	ja	nein	
<i>Qt Linguist .ts</i>	beides	ja	ja	nein	ja	ja ^{Seite 79, 10}	Bearbeitung erforderlich
<i>Android String-Ressourcen</i>	einsprachig	ja	ja ⁷	nein	nein	ja ^{Seite 79, 10}	

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Format	Linguali- tät ^{Seite 79, 1}	Pluralfor- men ²	Beschrei- bungen ³	Kontext ⁴	Ort ⁵	Markie- rungen ⁸	Zusätz- liche Status ⁶
<i>Apple iOS- Zeichenketter</i>	beides	nein	ja	nein	nein	nein	
<i>PHP- Zeichenketter</i>	einsprachig	nein ¹¹	ja	nein	nein	nein	
<i>JSON- Dateien</i>	einsprachig	nein	nein	nein	nein	nein	
<i>JSON- i18next- Dateien</i>	einsprachig	ja	nein	nein	nein	nein	
<i>go-i18n- JSON- Dateien</i>	einsprachig	ja	nein	nein	nein	nein	
<i>gotext- JSON- Dateien</i>	einsprachig	ja	ja	nein	ja	nein	
<i>ARB-Datei</i>	einsprachig	ja	ja	nein	nein	nein	
<i>WebExtension JSON</i>	einsprachig	ja	ja	nein	nein	nein	
<i>.XML- Ressourcen- Dateien</i>	einsprachig	nein	ja	nein	nein	ja ^{Seite 79, 10}	
<i>ResourceDict Dateien</i>	einsprachig	nein	nein	nein	nein	ja ^{Seite 79, 10}	
<i>CSV- Dateien</i>	beides	nein	ja	ja	ja	nein	Bear- beitung erforder- lich
<i>YAML- Dateien</i>	einsprachig	nein	ja	nein	nein	nein	
<i>Ruby YAML- Dateien</i>	einsprachig	ja	ja	nein	nein	nein	
<i>DTD- Dateien</i>	einsprachig	nein	nein	nein	nein	nein	
<i>Flat XML- Dateien</i>	einsprachig	nein	nein	nein	nein	ja ^{Seite 79, 10}	
<i>Windows RC- Dateien</i>	einsprachig	nein	ja	nein	nein	nein	
<i>Excel Open XML</i>	einsprachig	nein	ja	ja	ja	nein	Bear- beitung erforder- lich
<i>App-Store- Metadatenda</i>	einsprachig	nein	nein	nein	nein	nein	
<i>Untertitel- dateien</i>	einsprachig	nein	nein	nein	ja	nein	
<i>HTML- Dateien</i>	einsprachig	nein	nein	nein	nein	nein	
<i>OpenDocumε Format</i>	einsprachig	nein	nein	nein	nein	nein	
<i>IDML- Format</i>	einsprachig	nein	nein	nein	nein	nein	

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Format	Linguallität ^{Seite 79, 1}	Pluralformen ²	Beschreibungen ³	Kontext ⁴	Ort ⁵	Markierungen ⁸	Zusätzliche Status ⁶
<i>INI-Übersetzung</i>	einsprachig	nein	nein	nein	nein	nein	
<i>Inno Setup INI-Übersetzung</i>	einsprachig	nein	nein	nein	nein	nein	
<i>TermBase eXchange-Format</i>	zweisprachig	nein	ja	nein	nein	ja ^{Seite 79, 10}	
<i>Textdateien</i>	einsprachig	nein	nein	nein	nein	nein	
<i>Stringsdict-Format</i>	einsprachig	ja	ja	nein	nein	nein	
<i>Fluent-Format</i>	einsprachig	no ¹²	ja	nein	nein	nein	

Schreibgeschützte Zeichenketten

Neu in Version 3.10.

Read-only strings from translation files will be included, but can not be edited in Weblate. This feature is natively supported by few formats (*XLIFF* and *Android String-Ressourcen*), but can be emulated in others by adding a read-only flag, see *Anpassen des Verhaltens mit Markierungen*.

1.10.4 GNU gettext

Das am weitesten verbreitete Format für die Übersetzung von freier Software.

Contextual info stored in the file is supported by adjusting its headers or linking to corresponding source files.

The bilingual gettext PO file typically looks like this:

```
#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "Monday"
msgstr "Pondělí"

#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "Tuesday"
msgstr "Úterý"

#: weblate/accounts/avatar.py:163
msgctxt "No known user"
```

(Fortsetzung auf der nächsten Seite)

¹ Siehe *Zweisprachige und einsprachige Formate*

² Plurals are necessary to properly localize strings with variable count.

³ Source string descriptions can be used to pass additional info about the string to translate.

⁴ Context is used to differentiate identical strings used in different scopes (for example *Sun* can be used as an abbreviated name of the day „Sunday“ or as the name of our closest star).

⁵ Location of a string in source code might help proficient translators figure out how the string is used.

⁸ Siehe *Anpassen des Verhaltens mit Markierungen*

⁶ Additional states supported by the file format in addition to „Untranslated“ and „Translated“.

⁹ The gettext type comments are used as flags.

¹⁰ The flags are extracted from the non-standard attribute `weblate-flags` for all XML based formats. Additionally `max-length:N` is supported through the `maxwidth` attribute as defined in the XLIFF standard, see *Specifying translation flags*.

⁷ XML comment placed before the `<string>` element, parsed as a source string description.

¹¹ The plurals are supported only for Laravel which uses in string syntax to define them, see *Localization in Laravel*.

¹² Plurale werden in der Syntax der Zeichenketten behandelt und nicht als Plural in Weblate angezeigt.

(Fortsetzung der vorherigen Seite)

```
msgid "None"
msgstr "Žádný"
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	po/* .po
Einsprachige Basis-Sprachdatei	<i>Leer</i>
Vorlage für neue Übersetzungen	po/messages.pot
Dateiformat	<i>Gettext PO-Datei</i>

Siehe auch:

devel/gettext, devel/sphinx, [Gettext on Wikipedia](#), [PO Files](#), [Aktualisieren Sie die Variable ALL_LINGUAS in der „configure“-Datei](#), [Ausgabe von Gettext anpassen](#), [LINGUAS-Datei aktualisieren](#), [MO-Dateien erzeugen](#), [PO-Dateien auf POT aktualisieren \(msgmerge\)](#)

Monolingual gettext

Some projects decide to use gettext as monolingual formats—they code just the IDs in their source code and the string then needs to be translated to all languages, including English. This is supported, though you have to choose this file format explicitly when importing components into Weblate.

Die einsprachige gettext PO-Datei sieht normalerweise wie folgt aus:

```
#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-monday"
msgstr "Pondělí"

#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-tuesday"
msgstr "Úterý"

#: weblate/accounts/avatar.py:163
msgid "none-user"
msgstr "Žádný"
```

While the base language file will be:

```
#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-monday"
msgstr "Monday"

#: weblate/media/js/bootstrap-datepicker.js:1421
msgid "day-tuesday"
msgstr "Tuesday"

#: weblate/accounts/avatar.py:163
msgid "none-user"
msgstr "None"
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	po/* .po
Einsprachige Basis-Sprachdatei	po/en .po
Vorlage für neue Übersetzungen	po/messages.pot
Dateiformat	<i>Gettext PO-Datei (einsprachig)</i>

1.10.5 XLIFF

XML-based format created to standardize translation files, but in the end it is one of [many standards](#), in this area.

XML Localization Interchange File Format (XLIFF) is usually used as bilingual, but Weblate supports it as monolingual as well.

Weblate unterstützt XLIFF in mehreren Varianten:

XLIFF-Übersetzungsdatei

Simple XLIFF file where content of the elements is stored as plain text (all XML elements being escaped).

XLIFF mit Platzhalterunterstützung

Standard XLIFF supporting placeables and other XML elements.

XLIFF mit Gettext-Erweiterungen

XLIFF enriched by [XLIFF 1.2 Representation Guide for Gettext PO](#) to support plurals.

Siehe auch:

[XML Localization Interchange File Format \(XLIFF\) specification](#), [XLIFF 1.2 Representation Guide for Gettext PO](#)

Übersetzungsstatus

Geändert in Version 3.3: Weblate ignored the `state` attribute prior to the 3.3 release.

The `state` attribute in the file is partially processed and mapped to the „Needs edit“ state in Weblate (the following states are used to flag the string as needing edit if there is a target present: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). Should the `state` attribute be missing, a string is considered translated as soon as a `<target>` element exists.

If the translation string has `approved="yes"`, it will also be imported into Weblate as „Approved“, anything else will be imported as „Waiting for review“ (which matches the XLIFF specification).

While saving, Weblate doesn't add those attributes unless necessary:

- The `state` attribute is only added in case string is marked as needing edit.
- The `approved` attribute is only added in case string has been reviewed.
- In other cases the attributes are not added, but they are updated in case they are present.

That means that when using the XLIFF format, it is strongly recommended to turn on the Weblate review process, in order to see and change the approved state of strings.

Similarly upon importing such files (in the upload form), you should choose *Import as translated* under *Processing of strings needing edit*.

Siehe auch:

[Zugehörige Prüfer](#)

Leerzeichen und Zeilenumbrüche in XLIFF

Generally types or amounts of whitespace is not differentiated between in XML formats. If you want to keep it, you have to add the `xml:space="preserve"` flag to the string.

Zum Beispiel:

```
<trans-unit id="10" approved="yes">
  <source xml:space="preserve">hello</source>
  <target xml:space="preserve">Hello, world!
</target>
</trans-unit>
```

Specifying translation flags

You can specify additional translation flags (see *Anpassen des Verhaltens mit Markierungen*) by using the `weblate-flags` attribute. Weblate also understands `maxwidth` and `font` attributes from the XLIFF specification:

```
<trans-unit id="10" maxwidth="100" size-unit="pixel" font="ubuntu;22:bold">
  <source>Hello %s</source>
</trans-unit>
<trans-unit id="20" maxwidth="100" size-unit="char" weblate-flags="c-format">
  <source>Hello %s</source>
</trans-unit>
```

Das Attribut `font` wird nach Schriftfamilie, -größe und -gewicht analysiert, das obige Beispiel zeigt all das, obwohl nur die Schriftfamilie erforderlich ist. Jedes Leerzeichen in der Schriftfamilie wird in einen Unterstrich umgewandelt, so wird `Source Sans Pro` zu `Source_Sans_Pro`, bitte beachten Sie dies bei der Benennung der Schriftgruppe (siehe *Schriftarten verwalten*).

Zeichenkettenschlüssel

Weblate identifies the units in the XLIFF file by `resname` attribute in case it is present and falls back to `id` (together with `file` tag if present).

The `resname` attribute is supposed to be human friendly identifier of the unit making it more suitable for Weblate to display instead of `id`. The `resname` has to be unique in the whole XLIFF file. This is required by Weblate and is not covered by the XLIFF standard - it does not put any uniqueness restrictions on this attribute.

Typical Weblate <i>Konfigurationsdatei</i> for bilingual XLIFF	
Dateimaske	<code>localizations/*.xliif</code>
Einsprachige Basis-Sprachdatei	<i>Leer</i>
Vorlage für neue Übersetzungen	<code>localizations/en-US.xliif</code>
Dateiformat	<i>XLIFF-Übersetzungsdatei</i>

Typical Weblate <i>Konfigurationsdatei</i> for monolingual XLIFF	
Dateimaske	<code>localizations/*.xliif</code>
Einsprachige Basis-Sprachdatei	<code>localizations/en-US.xliif</code>
Vorlage für neue Übersetzungen	<code>localizations/en-US.xliif</code>
Dateiformat	<i>XLIFF-Übersetzungsdatei</i>

Siehe auch:

[XLIFF on Wikipedia](#), [XLIFF](#), [font attribute in XLIFF 1.2](#), [maxwidth attribute in XLIFF 1.2](#)

1.10.6 Java properties

Native Java format for translations.

Java properties are usually used as monolingual translations.

Weblate supports ISO-8859-1, UTF-8 and UTF-16 variants of this format. All of them support storing all Unicode characters, it is just differently encoded. In the ISO-8859-1, the Unicode escape sequences are used (for example `zkou\u0161ka`), all others encode characters directly either in UTF-8 or UTF-16.

Bemerkung: Loading escape sequences works in UTF-8 mode as well, so please be careful choosing the correct encoding set to match your application needs.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	src/app/Bundle_*.properties
Einsprachige Basis-Sprachdatei	src/app/Bundle.properties
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>Java Properties (ISO 8859-1)</i>

Siehe auch:

Java-Eigenschaften auf Wikipedia, Mozilla and Java properties files, *mi18n lang Dateien*, *GWT-Eigenschaften*, *updating-target-files*, *Die Übersetzung der Java-Eigenschaften-Datei formatieren*, *Übersetzungsdateien bereinigen*

1.10.7 mi18n lang Dateien

Neu in Version 4.7.

Dateiformat, das für die JavaScript-Lokalisierung von *mi18n* verwendet wird. Syntaktisch entspricht es *Java properties*.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	*.lang
Einsprachige Basis-Sprachdatei	en-US.lang
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>mi18n lang file</i>

Siehe auch:

mi18n, Mozilla and Java properties files, *Java properties*, *updating-target-files*, *Die Übersetzung der Java-Eigenschaften-Datei formatieren*, *Übersetzungsdateien bereinigen*

1.10.8 GWT-Eigenschaften

Natives GWT-Format für Übersetzungen.

GWT-Eigenschaften werden normalerweise als einsprachige Übersetzungen verwendet.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	src/app/Bundle_*.properties
Einsprachige Basis-Sprachdatei	src/app/Bundle.properties
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>GWT-Eigenschaften</i>

Siehe auch:

GWT-Lokalisierungsanleitung, GWT-Internationalisierungstutorial, Mozilla and Java properties files, *updating-target-files*, *Die Übersetzung der Java-Eigenschaften-Datei formatieren*, *Übersetzungsdateien bereinigen*

1.10.9 INI-Übersetzungen

Neu in Version 4.1.

INI-Dateiformat für Übersetzungen.

INI-Übersetzungen werden normalerweise als einsprachige Übersetzungen verwendet.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	language/*.ini
Einsprachige Basis-Sprachdatei	language/en.ini
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>INI-Datei</i>

Bemerkung: Weblate only extracts keys from sections within an INI file. In case your INI file lacks sections, you might want to use *Joomla-Übersetzungen* or *Java properties* instead.

Siehe auch:

INI Files, *Java properties*, *Joomla-Übersetzungen*, *Inno Setup INI-Übersetzungen*

1.10.10 Inno Setup INI-Übersetzungen

Neu in Version 4.1.

Inno Setup INI-Dateiformat für Übersetzungen.

Inno Setup INI-Übersetzungen werden normalerweise als einsprachige Übersetzungen verwendet.

Bemerkung: The only notable difference to *INI-Übersetzungen* is in supporting %n and %t placeholders for line break and tab.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	language/*.isl
Einsprachige Basis-Sprachdatei	language/en.isl
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>Inno Setup INI-Datei</i>

Bemerkung: Only Unicode files (.isl) are currently supported, ANSI variant (.isl) is currently not supported.

Siehe auch:

INI Files, *Joomla-Übersetzungen*, *INI-Übersetzungen*

1.10.11 Joomla-Übersetzungen

Neu in Version 2.12.

Natives Joomla-Format für Übersetzungen.

Joomla-Übersetzungen werden normalerweise als einsprachige Übersetzungen verwendet.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	language/*/com_foobar.ini
Einsprachige Basis-Sprachdatei	language/en-GB/com_foobar.ini
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>Joomla-Sprachdatei</i>

Siehe auch:

Mozilla and Java properties files, *INI-Übersetzungen*, *Inno Setup INI-Übersetzungen*

1.10.12 Qt Linguist .ts

In Qt-basierten Anwendungen verwendetes Übersetzungsformat.

Qt Linguist-Dateien werden sowohl als zweisprachige als auch als einsprachige Übersetzungen verwendet.

Typical Weblate <i>Komponentenkonfiguration</i> when using as bilingual	
Dateimaske	i18n/app.*.ts
Einsprachige Basis-Sprachdatei	<i>Leer</i>
Vorlage für neue Übersetzungen	i18n/app.de.ts
Dateiformat	<i>Qt Linguist-Übersetzungsdatei</i>

Typical Weblate <i>Komponentenkonfiguration</i> when using as monolingual	
Dateimaske	i18n/app.*.ts
Einsprachige Basis-Sprachdatei	i18n/app.en.ts
Vorlage für neue Übersetzungen	i18n/app.en.ts
Dateiformat	<i>Qt Linguist-Übersetzungsdatei</i>

Siehe auch:

Qt-Linguist-Handbuch, Qt .ts, *Zweisprachige und einsprachige Formate*

1.10.13 Android String-Ressourcen

Android-spezifisches Dateiformat zum Übersetzen von Anwendungen.

Android string resources are monolingual, the *Einsprachige Basis-Sprachdatei* is stored in a different location from the other files – `res/values/strings.xml`.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	res/values-*/strings.xml
Einsprachige Basis-Sprachdatei	res/values/strings.xml
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>Android String-Ressource</i>

Siehe auch:

[Android string resources documentation](#), [Android string resources](#)

Bemerkung: Android *string-array* structures are not currently supported. To work around this, you can break your string arrays apart:

```
<string-array name="several_strings">
  <item>First string</item>
  <item>Second string</item>
</string-array>
```

become:

```
<string-array name="several_strings">
  <item>@string/several_strings_0</item>
  <item>@string/several_strings_1</item>
</string-array>
<string name="several_strings_0">First string</string>
<string name="several_strings_1">Second string</string>
```

The *string-array* that points to the *string* elements should be stored in a different file, and not be made available for translation.

This script may help pre-process your existing strings.xml files and translations: <https://gist.github.com/paour/11291062>

Hinweis: To avoid translating some strings, these can be marked as non-translatable. This can be especially useful for string references:

```
<string name="foobar" translatable="false">@string/foo</string>
```

1.10.14 Apple iOS-Zeichenketten

File format typically used for translating Apple iOS applications, but also standardized by PWG 5100.13 and used on NeXTSTEP/OpenSTEP.

Apple iOS-Zeichenketten werden normalerweise einsprachig verwendet.

Typical Weblate <i>Konfigurationsdateien</i>			
Dateimaske			Resources/*.lproj/Localizable.strings
Einsprachige Basis-Sprachdatei			Resources/en.lproj/Localizable.strings oder Resources/Base.lproj/Localizable.strings
Vorlage für neue Übersetzungen			<i>Leer</i>
Dateiformat			<i>iOS-Zeichenketten (UTF-8)</i>

Siehe auch:

[Stringsdict-Format](#), [Apple „Strings Files“-Dokumentation](#), [Message Catalog File Format in PWG 5100.13](#), [Mac OSX strings](#)

1.10.15 PHP-Zeichenketten

PHP-Übersetzungen sind normalerweise einsprachig, daher ist es empfehlenswert, eine Basisdatei mit (meist) englischen Zeichenketten anzugeben.

Beispieldatei:

```
<?php
$LANG['foo'] = 'bar';
$LANG['foo1'] = 'foo bar';
$LANG['foo2'] = 'foo bar baz';
$LANG['foo3'] = 'foo bar baz bag';
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	lang/*/texts.php
Einsprachige Basis-Sprachdatei	lang/en/texts.php
Vorlage für neue Übersetzungen	lang/en/texts.php
Dateiformat	<i>PHP-Zeichenketten</i>

Laravel PHP-Zeichenketten

Geändert in Version 4.1.

Die Laravel-PHP-Lokalisierungsdateien mit Pluralformen werden auch unterstützt:

```
<?php
return [
    'welcome' => 'Welcome to our application',
    'apples' => 'There is one apple|There are many apples',
];
```

Siehe auch:

PHP, Lokalisierung in Laravel

1.10.16 JSON-Dateien

Neu in Version 2.0.

Geändert in Version 2.16: Seit Weblate 2.16 und mit [translate-toolkit](#) mindestens 2.2.4 werden auch JSON-Nested-Structure-Dateien unterstützt.

Geändert in Version 4.3: The structure of JSON file is properly preserved even for complex situations which were broken in prior releases.

JSON format is used mostly for translating applications implemented in JavaScript.

Weblate unterstützt derzeit mehrere Varianten von JSON-Übersetzungen:

- Simple key / value files, used for example by *vue-i18n* or *react-intl*.
- Files with nested keys.
- *JSON-i18next-Dateien*
- *go-i18n-JSON-Dateien*
- *gotext-JSON-Dateien*
- *WebExtension-JSON*
- *ARB-Datei*

JSON-Übersetzungen sind normalerweise einsprachig, daher ist es empfehlenswert, eine Basisdatei mit (meist) englischen Zeichenketten anzugeben.

Beispieldatei:

```
{
  "Hello, world!\n": "Ahoj světe!\n",
  "Orangutan has %d banana.\n": "",
  "Try Weblate at https://demo.weblate.org/!\n": "",
  "Thank you for using Weblate.": ""
}
```

Nested files are supported as well (see above for requirements), such a file can look like:

```
{
  "weblate": {
    "hello": "Ahoj světe!\n",
    "orangutan": "",
    "try": "",
    "thanks": ""
  }
}
```

Hinweis: The *JSON file* and *JSON nested structure file* can both handle same type of files. Both preserve existing JSON structure when translating.

The only difference between them is when adding new strings using Weblate. The nested structure format parses the newly added key and inserts the new string into the matching structure. For example `app.name` key is inserted as:

```
{
  "app": {
    "name": "Weblate"
  }
}
```

Typical Weblate <i>Komponentenkonfiguration</i>	
---	--

Dateimaske	langs/translation-*.json
Einsprachige Basis-Sprachdatei	langs/translation-en.json
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>JSON-Nested-Structure-Datei</i>

Siehe auch:

[JSON](#), [updating-target-files](#), [Ausgabe von JSON anpassen](#), [Übersetzungsdateien bereinigen](#),

1.10.17 JSON-i18next-Dateien

Geändert in Version 2.17: Seit Weblate 2.17 und mit [translate-toolkit](#) mindestens 2.2.5 werden auch i18next JSON-Dateien mit Pluralformen unterstützt.

Geändert in Version 4.15.1: Unterstützung für die v4-Variante dieses Formats hinzugefügt.

Hinweis: In case you use plurals, it is recommended to use v4 as that aligned plural handling with CLDR. Older versions have different plural rules for some languages which are not correct.

[i18next](#) is an internationalization framework written in and for JavaScript. Weblate supports its localization files with features such as plurals.

i18next-Übersetzungen sind einsprachig, daher ist es empfehlenswert, eine Basisdatei mit (meist) englischen Zeichenketten anzugeben.

Bemerkung: Weblate unterstützt die Varianten i18next JSON v3 und v4. Bitte wählen Sie das richtige Dateiformat für Ihre Einsatzumgebung.

Die Varianten v2 und v1 sind größtenteils kompatibel mit v3, mit Ausnahme der Behandlung von Pluralformen.

Beispieldatei:

```
{
  "hello": "Hello",
  "apple": "I have an apple",
  "apple_plural": "I have {{count}} apples",
  "apple_negative": "I have no apples"
}
```

Typical Weblate <i>Konfigurationsdatei</i>	
Dateimaske	langs/*.json
Einsprachige Basis-Sprachdatei	langs/en.json
Vorlage für neue Übersetzungen	Leer
Dateiformat	i18next-JSON-Datei v3

Siehe auch:

[JSON](#), [i18next JSON-Format](#), [updating-target-files](#), [Ausgabe von JSON anpassen](#), [Übersetzungsdateien bereinigen](#)

1.10.18 go-i18n-JSON-Dateien

Neu in Version 4.1.

go-i18n-Übersetzungen sind einsprachig, daher ist es empfehlenswert, eine Basisdatei mit (meist) englischen Zeichenketten anzugeben.

Geändert in Version 4.16: Weblate unterstützt sowohl das v1- als auch das v2-Format, bitte bei der Auswahl des Dateiformats die entsprechende Version auswählen.

Typical Weblate <i>Konfigurationsdatei</i>	
Dateimaske	langs/*.json
Einsprachige Basis-Sprachdatei	langs/en.json
Vorlage für neue Übersetzungen	Leer
Dateiformat	go-i18n-JSON-Datei

Siehe auch:

[JSON](#), [go-i18n](#), [updating-target-files](#), [Ausgabe von JSON anpassen](#), [Übersetzungsdateien bereinigen](#),

1.10.19 gotext-JSON-Dateien

Neu in Version 4.15.1.

gotext-Übersetzungen sind einsprachig, daher ist es empfehlenswert, eine Basisdatei mit (meist) englischen Zeichenketten anzugeben.

Typical Weblate <i>Komponentenkonfiguration</i>		
Dateimaske		internal/translations/locales/*/messages.gotext.json
Einsprachige Sprachdatei	Basis-	internal/translations/locales/en-GB/messages.gotext.json
Vorlage für neue Übersetzungen		<i>Leer</i>
Dateiformat		<i>gotext-JSON-Datei</i>

Siehe auch:

[JSON](#), [I18n in Go: Managing Translations](#), [updating-target-files](#), [Ausgabe von JSON anpassen](#), [Übersetzungsdateien bereinigen](#),

1.10.20 ARB-Datei

Neu in Version 4.1.

ARB-Übersetzungen sind einsprachig, daher ist es empfehlenswert, eine Basisdatei mit (meist) englischen Zeichenketten anzugeben.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	lib/l10n/intl_*.arb
Einsprachige Basis-Sprachdatei	lib/l10n/intl_en.arb
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>ARB-Datei</i>

Siehe auch:

[JSON](#), [Application Resource Bundle Specification](#), [Internationalizing Flutter apps](#), [updating-target-files](#), [Ausgabe von JSON anpassen](#), [Übersetzungsdateien bereinigen](#)

1.10.21 WebExtension-JSON

Neu in Version 2.16: Dies wird seit Weblate 2.16 und mit [translate-toolkit](#) mindestens 2.2.4 unterstützt.

File format used when translating extensions for Mozilla Firefox or Google Chromium.

Bemerkung: While this format is called JSON, its specification allows to include comments, which are not part of JSON specification. Weblate currently does not support file with comments.

Beispieldatei:

```
{
  "hello": {
    "message": "Ahoj světe!\n",
    "description": "Description",
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

"placeholders": {
  "url": {
    "content": "$1",
    "example": "https://developer.mozilla.org"
  }
},
"orangutan": {
  "message": "Orangutan has $coUnT$ bananas",
  "description": "Description",
  "placeholders": {
    "count": {
      "content": "$1",
      "example": "5"
    }
  }
},
"try": {
  "message": "",
  "description": "Description"
},
"thanks": {
  "message": "",
  "description": "Description"
}
}

```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	<code>_locales/*/messages.json</code>
Einsprachige Basis-Sprachdatei	<code>_locales/en/messages.json</code>
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>WebExtension-JSON-Datei</i>

Siehe auch:

JSON, [Google chrome.i18n](#), [Mozilla Extensions Internationalization](#)

1.10.22 .XML-Ressourcen-Dateien

Neu in Version 2.3.

A .XML resource (.resx) file employs a monolingual XML file format used in Microsoft .NET applications. It is interchangeable with .resw, when using identical syntax to .resx.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	<code>Resources/Language.*.resx</code>
Einsprachige Basis-Sprachdatei	<code>Resources/Language.resx</code>
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>.NET-Ressourcen-Datei</i>

Siehe auch:

[.NET Resource files \(.resx\)](#), [updating-target-files](#), [Übersetzungsdateien bereinigen](#)

1.10.23 ResourceDictionary-Dateien

Neu in Version 4.13.

ResourceDictionary is a monolingual XML file format used to package localizable string resources for Windows Presentation Foundation (WPF) applications.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	Languages/* .xaml
Einsprachige Basis-Sprachdatei	Language/en .xaml
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>ResourceDictionary-Datei</i>

Siehe auch:

[Flat XML](#), [Flat XML-Dateien](#), [updating-target-files](#), [Übersetzungsdateien bereinigen](#)

1.10.24 CSV-Dateien

Neu in Version 2.4.

CSV files can contain a simple list of source and translation. Weblate supports the following files:

- Files with header defining fields (location, source, target, ID, fuzzy, context, translator_comments, developer_comments). This is the recommended approach, as it is the least error prone. Choose *CSV file* as a file format.
- Files with two fields—source and translation (in this order). Choose *Simple CSV file* as a file format.
- Headerless files with fields in order defined by the [translate-toolkit](#): location, source, target, ID, fuzzy, context, translator_comments, developer_comments. Choose *CSV file* as a file format.
- Remember to define *Einsprachige Basis-Sprachdatei* when your files are monolingual (see *Zweisprachige und einsprachige Formate*).

Hinweis: Standardmäßig erkennt das CSV-Format die Dateikodierung automatisch. Dies kann in einigen Ausnahmefällen unzuverlässig sein und zu Leistungseinbußen führen. Bitte wählen Sie eine Dateiformatvariante mit Kodierung, um dies zu vermeiden (zum Beispiel *CSV-Datei (UTF-8)*).

Warnung: Das CSV-Format erkennt derzeit automatisch den Dialekt der CSV-Datei. In einigen Fällen kann die automatische Erkennung fehlschlagen und Sie erhalten gemischte Ergebnisse. Dies gilt insbesondere für CSV-Dateien mit Zeilenumbrüchen in den Werten. Als Abhilfe empfiehlt es sich, die Anführungszeichen wegzulassen.

Beispieldatei:

```
Thank you for using Weblate.,Děkujeme za použití Weblate.
```

Typical Weblate <i>Komponentenkonfiguration</i> for bilingual CSV	
Dateimaske	locale/* .csv
Einsprachige Basis-Sprachdatei	<i>Leer</i>
Vorlage für neue Übersetzungen	locale/en .csv
Dateiformat	<i>CSV-Datei</i>

Typical Weblate <i>Komponentenkonfiguration</i> for monolingual CSV	
Dateimaske	locale/*.csv
Einsprachige Basis-Sprachdatei	locale/en.csv
Vorlage für neue Übersetzungen	locale/en.csv
Dateiformat	<i>Einfache CSV-Datei</i>

Mehrwertige CSV-Datei (UTF-8)

Neu in Version 4.13.

This variant of the CSV files allows storing multiple translations per string.

Siehe auch:

[CSV](#)

1.10.25 YAML-Dateien

Neu in Version 2.9.

The plain YAML files with string keys and values. Weblate also extract strings from lists or dictionaries.

Beispiel für eine YAML-Datei:

```
weblate:
  hello: ""
  orangutan: ""
  try: ""
  thanks: ""
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	translations/messages/*.yaml
Einsprachige Basis-Sprachdatei	translations/messages.en.yaml
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>YAML-Datei</i>

Siehe auch:

[YAML](#), [Ruby YAML-Dateien](#)

1.10.26 Ruby YAML-Dateien

Neu in Version 2.9.

Ruby i18n YAML files with language as root node.

Beispiel für eine Ruby i18n YAML-Datei:

```
cs:
  weblate:
    hello: ""
    orangutan: ""
    try: ""
    thanks: ""
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	translations/messages.*.yaml
Einsprachige Basis-Sprachdatei	translations/messages.en.yaml
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>Ruby YAML-Datei</i>

Siehe auch:

[YAML](#), [YAML-Dateien](#)

1.10.27 DTD-Dateien

Neu in Version 2.18.

Beispiel für eine DTD-Datei:

```
<!ENTITY hello "">
<!ENTITY orangutan "">
<!ENTITY try "">
<!ENTITY thanks "">
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	locale/*.dtd
Einsprachige Basis-Sprachdatei	locale/en.dtd
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>DTD-Datei</i>

Siehe auch:

[Mozilla DTD format](#)

1.10.28 Flat XML-Dateien

Neu in Version 3.9.

Beispiel für eine Flat XML-Datei:

```
<?xml version='1.0' encoding='UTF-8'?>
<root>
  <str key="hello_world">Hello World!</str>
  <str key="resource_key">Translated value.</str>
</root>
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	locale/*.xml
Einsprachige Basis-Sprachdatei	locale/en.xml
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>Flat XML-Datei</i>

Siehe auch:

[Flat XML](#)

1.10.29 Windows RC-Dateien

Geändert in Version 4.1: Unterstützung für Windows RC-Dateien neu geschrieben.

Bemerkung: Die Unterstützung für dieses Format befindet sich derzeit in der Beta-Phase, Rückmeldungen aus der Testphase sind willkommen.

Beispiel für eine Windows RC-Datei:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

STRINGTABLE
BEGIN
    IDS_MSG1            "Hello, world!\n"
    IDS_MSG2            "Orangutan has %d banana.\n"
    IDS_MSG3            "Try Weblate at http://demo.weblate.org/!\n"
    IDS_MSG4            "Thank you for using Weblate."
END
```

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	lang/* .rc
Einsprachige Basis-Sprachdatei	lang/en-US.rc
Vorlage für neue Übersetzungen	lang/en-US.rc
Dateiformat	<i>RC-Datei</i>

Siehe auch:

[Windows RC files](#)

1.10.30 App-Store-Metadatendateien

Neu in Version 3.5.

Metadata used for publishing apps in various app stores can be translated. Currently the following tools are compatible:

- [Triple-T gradle-play-publisher](#)
- [Fastlane](#)
- [F-Droid](#)

The metadata consists of several textfiles, which Weblate will present as separate strings to translate.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	fastlane/android/metadata/*
Einsprachige Basis-Sprachdatei	fastlane/android/metadata/en-US
Vorlage für neue Übersetzungen	fastlane/android/metadata/en-US
Dateiformat	<i>App-Store-Metadaten-Dateien</i>

Hinweis: Falls Sie bestimmte Zeichenketten (z. B. Änderungsprotokolle) nicht übersetzen wollen, markieren Sie diese als schreibgeschützt (siehe [Anpassen des Verhaltens mit Markierungen](#)). Dies kann durch die [Massenbearbeitung](#) automatisiert werden.

1.10.31 Untertiteldateien

Neu in Version 3.7.

Weblate kann verschiedene Untertiteldateien übersetzen:

- SubRip-Untertiteldatei (* .srt)
- MicroDVD-Untertiteldatei (* .sub)
- Advanced Substation Alpha-Untertiteldatei (* .ass)
- Substation Alpha-Untertiteldatei (* .ssa)

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	path/* .srt
Einsprachige Basis-Sprachdatei	path/en .srt
Vorlage für neue Übersetzungen	path/en .srt
Dateiformat	<i>SubRip-Untertiteldatei</i>

Siehe auch:

[Subtitles](#)

1.10.32 Excel Open XML

Neu in Version 3.2.

Excel Open XML (.xlsx) files can be imported and exported.

When uploading XLSX files for translation, be aware that only the active worksheet is considered, and there must be at least a column called `source` (which contains the source string) and a column called `target` (which contains the translation). Additionally there should be the column called `context` (which contains the context path of the translation string). If you use the XLSX download for exporting the translations into an Excel workbook, you already get a file with the correct file format.

1.10.33 HTML-Dateien

Neu in Version 4.1.

Bemerkung: Die Unterstützung für dieses Format befindet sich derzeit in der Beta-Phase, Rückmeldungen aus der Testphase sind willkommen.

The translatable content is extracted from the HTML files and offered for the translation.

Siehe auch:

[HTML](#)

1.10.34 Textdateien

Neu in Version 4.6.

Bemerkung: Die Unterstützung für dieses Format befindet sich derzeit in der Beta-Phase, Rückmeldungen aus der Testphase sind willkommen.

The translatable content is extracted from the plain text files and offered for the translation. Each paragraph is translated as a separate string.

Dieses Format gibt es in drei Varianten:

- Klartextdatei
- DokuWiki-Textdatei
- MediaWiki-Textdatei

Siehe auch:

[Simple Text Documents](#)

1.10.35 OpenDocument-Format

Neu in Version 4.1.

Bemerkung: Die Unterstützung für dieses Format befindet sich derzeit in der Beta-Phase, Rückmeldungen aus der Testphase sind willkommen.

The translatable content is extracted from the OpenDocument files and offered for the translation.

Siehe auch:

[OpenDocument Format](#)

1.10.36 IDML-Format

Neu in Version 4.1.

Bemerkung: Die Unterstützung für dieses Format befindet sich derzeit in der Beta-Phase, Rückmeldungen aus der Testphase sind willkommen.

The translatable content is extracted from the Adobe InDesign Markup Language files and offered for the translation.

1.10.37 TermBase eXchange-Format

Neu in Version 4.5.

TBX is an XML format for the exchange of terminology data.

Typical Weblate <i>Komponentenkonfiguration</i>	
Dateimaske	tbx/*.*tbx
Einsprachige Basis-Sprachdatei	<i>Leer</i>
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>TermBase eXchange-Format</i>

Siehe auch:

[TBX auf Wikipedia](#), [TBX](#), [Glossar](#)

1.10.38 Stringsdict-Format

Neu in Version 4.8.

Bemerkung: Die Unterstützung für dieses Format befindet sich derzeit in der Beta-Phase, Rückmeldungen aus der Testphase sind willkommen.

Ein von Apple verwendetes XML-basiertes Format, das mehrere Pluralformen einer Zeichenkette speichern kann.

Typical Weblate <i>Konfigurationsdatei</i>			
Dateimaske		Resources/*. <i>lproj</i> /Localizable.stringsdict	
Einsprachige Basis-Sprachdatei		Resources/en. <i>lproj</i> /Localizable.stringsdict	oder Resources/Base. <i>lproj</i> /Localizable.stringsdict
Vorlage für neue Übersetzungen		<i>Leer</i>	
Dateiformat		<i>Stringsdict-Datei</i>	

Siehe auch:

[Apple iOS-Zeichenketten](#), [Stringsdict-Dateiformat](#)

1.10.39 Fluent-Format

Neu in Version 4.8.

Bemerkung: Die Unterstützung für dieses Format befindet sich derzeit in der Beta-Phase, Rückmeldungen aus der Testphase sind willkommen.

Fluent ist ein einsprachiges Textformat, das sich auf asymmetrische Lokalisierung konzentriert: Eine einfache Zeichenkette in einer Sprache kann in einer anderen Sprache eine komplexe Übersetzung mit mehreren Varianten ergeben.

Typical Weblate <i>Konfigurationsdatei</i>	
Dateimaske	locales/*/messages.ftl
Einsprachige Basis-Sprachdatei	locales/en/messages.ftl
Vorlage für neue Übersetzungen	<i>Leer</i>
Dateiformat	<i>Fluent-Datei</i>

Siehe auch:

[Projekt Fluent Website](#)

1.10.40 Unterstützung anderer Formate

Most formats supported by [translate-toolkit](#) which support serializing can be easily supported, but they did not (yet) receive any testing. In most cases some thin layer is needed in Weblate to hide differences in behavior of different [translate-toolkit](#) storages.

To add support for a new format, the preferred approach is to first implement support for it in the [translate-toolkit](#).

Siehe auch:

[Translation Related File Formats](#)

1.11 Integration der Versionsverwaltung

Weblate currently supports *Git* (with extended support for *GitHub-Pull-Requests*, *GitLab-Merge-Requests*, *Gitea-Pull-Requests*, *Gerrit*, *Subversion* and *Bitbucket Server pull requests*) and *Mercurial* as version control back-ends.

1.11.1 Zugriff auf Repositorys

The VCS repository you want to use has to be accessible to Weblate. With a publicly available repository you just need to enter the correct URL (for example `https://github.com/WeblateOrg/weblate.git`), but for private repositories or for push URLs the setup is more complex and requires authentication.

Zugriff auf Repositorys von Hosted Weblate aus

For Hosted Weblate there is a dedicated push user registered on GitHub, Bitbucket, Codeberg and GitLab (with the username *weblate*, e-mail `hosted@weblate.org` and, named *Weblate push user*). You need to add this user as a collaborator and give it appropriate permission to your repository (read-only is okay for cloning, write is required for pushing). Depending on service and your organization settings, this happens immediately, or requires confirmation on the Weblate side.

The *weblate* user on GitHub accepts invitations automatically within five minutes. Manual processing might be needed on the other services, so please be patient.

Once the *weblate* user is added, you can configure *Quellcode-Repository* and *Push-URL für Repository* using the SSH protocol (for example `git@github.com:WeblateOrg/weblate.git`).

SSH-Repositorys

The most frequently used method to access private repositories is based on SSH. Authorize the public Weblate SSH key (see [Weblate-SSH-Schlüssel](#)) to access the upstream repository this way.

Warnung: On GitHub, each key can only be used once, see [GitHub-Repositorys](#) and [Zugriff auf Repositorys von Hosted Weblate aus](#).

Weblate also stores the host key fingerprint upon first connection, and fails to connect to the host should it be changed later (see [Verifying SSH host keys](#)).

Falls eine Anpassung erforderlich ist, nehmen Sie diese über die Weblate-Adminoberfläche vor:

Der öffentliche Schlüssel von Weblate ist für alle Benutzer sichtbar, welche die Seite *Über Weblate* besuchen.

Admins can generate or display the public key currently used by Weblate in the connection (from *SSH keys*) on the admin interface landing page.

Bemerkung: The corresponding private SSH key can not currently have a password, so make sure it is well protected.

Hinweis: Make a backup of the generated private Weblate SSH key.

Verifying SSH host keys

Weblate speichert die SSH-Hostschlüssel beim ersten Zugriff automatisch und merkt sie sich für die weitere Verwendung.

In case you want to verify the key fingerprint before connecting to the repository, add the SSH host keys of the servers you are going to access in *Add host key*, from the same section of the admin interface. Enter the hostname you are going to access (e.g. `gitlab.com`), and press *Submit*. Verify its fingerprint matches the server you added.

The added keys with fingerprints are shown in the confirmation message:

Manage / SSH keys

Added host key for github.com with fingerprint p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM (ecdsa-sha2-nistp256), please verify that it is correct.

Added host key for github.com with fingerprint nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8 (ssh-rsa), please verify that it is correct.

Added host key for github.com with fingerprint +DiY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU (ssh-ed25519), please verify that it is correct.

Weblate status Backups Translation memory Performance report **SSH keys** Alerts Repositories Users Teams

Appearance Tools Automatic suggestions Billing

Public SSH key ⓘ

Weblate uses SSH key to access remote repositories. The corresponding public key is found below, you can use it to grant Weblate access to a repository. ⓘ

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNz8fKPCbs6hiYpnovamGbWDxygRSjmbGwjV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhLwlp1uYlW
Weblate
```

Download private key

Known host keys ⓘ

Hostname	Key type	Fingerprint
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-ed25519	+DiY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ssh-rsa	nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8

Add host key ⓘ

To access SSH hosts, its host key needs to be verified. You can get the host key by entering a domain name or IP for the host in the form below.

Hostname Port

Submit

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

GitHub-Repositorys

Der Zugriff über SSH ist möglich (siehe *SSH-Repositorys*), aber falls Sie auf mehr als ein Repository zugreifen müssen, stoßen Sie auf eine GitHub-Beschränkung für die Verwendung von SSH-Schlüsseln (da jeder Schlüssel nur einmal verwendet werden kann).

In case the *Push Branch* is not set, the project is forked and changes pushed through a fork. In case it is set, changes are pushed to the upstream repository and chosen branch.

For smaller deployments, use HTTPS authentication with a personal access token and your GitHub account, see *Creating an access token for command-line use*.

For bigger setups, it is usually better to create a dedicated user for Weblate, assign it the public SSH key generated in Weblate (see *Weblate-SSH-Schlüssel*) and grant it access to all the repositories you want to translate. This approach is also used for Hosted Weblate, there is dedicated *weblate* user for that.

Siehe auch:

Zugriff auf Repositorys von Hosted Weblate aus

Weblate-interne URLs

Teilen Sie ein Repository-Setup zwischen verschiedenen Komponenten, indem Sie auf seine Platzierung als `weblate://project/component` in anderen (verknüpften) Komponenten verweisen. Auf diese Weise verwenden verknüpfte Komponenten die VCS-Repository-Konfiguration der Haupt(referenzierten) Komponente.

Warnung: Beim Entfernen der Hauptkomponente werden auch die verknüpften Komponenten entfernt.

Weblate automatically adjusts the repository URL when creating a component if it finds a component with a matching repository setup. You can override this in the last step of the component configuration.

Reasons to use this:

- Saves disk space on the server, the repository is stored just once.
- Makes the updates faster, only one repository is updated.
- There is just single exported repository with Weblate translations (see *Git-Exporter*).
- Einige Erweiterungen können mit mehreren Komponenten arbeiten, die sich ein Repository teilen, zum Beispiel *Git-Commits zusammenfassen*.

HTTPS-Repositorys

To access protected HTTPS repositories, include the username and password in the URL. Don't worry, Weblate will strip this info when the URL is shown to users (if even allowed to see the repository URL at all).

For example the GitHub URL with authentication added might look like: `https://user:your_access_token@github.com/WeblateOrg/weblate.git`.

Bemerkung: Wenn Ihr Benutzername oder Ihr Passwort Sonderzeichen enthält, müssen diese in der URL kodiert werden, zum Beispiel `https://user%40example.com:%24password%23@bitbucket.org/...`

Verwendung eines Proxys

If you need to access HTTP/HTTPS VCS repositories using a proxy server, configure the VCS to use it.

This can be done using the `http_proxy`, `https_proxy`, and `all_proxy` environment variables, (as described in the [cURL documentation](#)) or by enforcing it in the VCS configuration, for example:

```
git config --global http.proxy http://user:password@proxy.example.com:80
```

Bemerkung: The proxy configuration needs to be done under user running Weblate (see also [Dateisystemberechtigungen](#)) and with `HOME=$DATA_DIR/home` (see [DATA_DIR](#)), otherwise Git executed by Weblate will not use it.

Siehe auch:

The [cURL manpage](#), [Git config documentation](#)

1.11.2 Git

Hinweis: Weblate benötigt Git 2.12 oder neuer.

Siehe auch:

See [Zugriff auf Repositories](#) for info on how to access different kinds of repositories.

Git Push erzwingen

This behaves exactly like Git itself, the only difference being that it always force pushes. This is intended only in the case of using a separate repository for translations.

Warnung: Use with caution, as this easily leads to lost commits in your upstream repository.

Customizing Git configuration

Weblate invokes all VCS commands with `HOME=$DATA_DIR/home` (see [DATA_DIR](#)), therefore editing the user configuration needs to be done in `DATA_DIR/home/.git`.

Git remote helpers

You can also use Git [remote helpers](#) for additionally supporting other version control systems, but be prepared to debug problems this may lead to.

At this time, helpers for Bazaar and Mercurial are available within separate repositories on GitHub: [git-remote-hg](#) and [git-remote-bzr](#). Download them manually and put somewhere in your search path (for example `~/bin`). Make sure you have the corresponding version control systems installed.

Once you have these installed, such remotes can be used to specify a repository in Weblate.

To clone the `gnuhello` project from Launchpad using Bazaar:

```
bzr::lp:gnuhello
```

For the `hello` repository from `selenic.com` using Mercurial:

```
hg::http://selenic.com/repo/hello
```

Warnung: The inconvenience of using Git remote helpers is for example with Mercurial, the remote helper sometimes creates a new tip when pushing changes back.

1.11.3 GitHub-Pull-Requests

Neu in Version 2.3.

This adds a thin layer atop *Git* using the *GitHub API* to allow pushing translation changes as pull requests, instead of pushing directly to the repository.

Git pusht Änderungen direkt in ein Repository, während *GitHub-Pull-Requests* Pull Requests erstellt. Letzteres wird für den bloßen Zugriff auf Git-Repositorys nicht benötigt.

You need to configure API credentials (*GITHUB_CREDENTIALS*) in the Weblate settings to make this work. Once configured, you will see a *GitHub* option when selecting *Versionsverwaltung*.

Siehe auch:

Pushen von Änderungen aus Weblate, GITHUB_CREDENTIALS

1.11.4 GitLab-Merge-Requests

Neu in Version 3.9.

This just adds a thin layer atop *Git* using the *GitLab API* to allow pushing translation changes as merge requests instead of pushing directly to the repository.

Es besteht keine Notwendigkeit, dies für den Zugriff auf Git-Repositorys zu verwenden, das gewöhnliche *Git* funktioniert genauso, der einzige Unterschied ist, wie das Pushen zu einem Repository gehandhabt wird. Mit *Git* werden Änderungen direkt in das Repository gepusht, während *GitLab-Merge-Requests* einen Merge Request erstellt.

You need to configure API credentials (*GITLAB_CREDENTIALS*) in the Weblate settings to make this work. Once configured, you will see a *GitLab* option when selecting *Versionsverwaltung*.

Siehe auch:

Pushen von Änderungen aus Weblate, GITLAB_CREDENTIALS

1.11.5 Gitea-Pull-Requests

Neu in Version 4.12.

This just adds a thin layer atop *Git* using the *Gitea API* to allow pushing translation changes as pull requests instead of pushing directly to the repository.

Es besteht keine Notwendigkeit, dies für den Zugriff auf Git-Repositorys zu verwenden, das gewöhnliche *Git* funktioniert genauso, der einzige Unterschied ist, wie das Pushen zu einem Repository gehandhabt wird. Mit *Git* werden Änderungen direkt in das Repository gepusht, während *Gitea-Pull-Requests* Pull Requests erstellt.

Damit dies funktioniert, müssen Sie die API-Anmeldeinformationen (*GITEA_CREDENTIALS*) in den Weblate-Einstellungen konfigurieren. Nach der Konfiguration sehen Sie die Option *Gitea*, wenn Sie *Versionsverwaltung* auswählen.

Siehe auch:

Pushen von Änderungen aus Weblate, GITEA_CREDENTIALS

1.11.6 Bitbucket Server pull requests

Neu in Version 4.16.

This just adds a thin layer atop [Git](#) using the [Bitbucket Server API](#) to allow pushing translation changes as pull requests instead of pushing directly to the repository.

Warnung: This does not support Bitbucket Cloud API.

There is no need to use this to access Git repositories, ordinary [Git](#) works the same, the only difference is how pushing to a repository is handled. With [Git](#) changes are pushed directly to the repository, while [Bitbucket Server pull requests](#) creates pull request.

You need to configure API credentials ([BITBUCKETSERVER_CREDENTIALS](#)) in the Weblate settings to make this work. Once configured, you will see a [Bitbucket Server](#) option when selecting [Versionsverwaltung](#).

Siehe auch:

Pushen von Änderungen aus Weblate, [BITBUCKETSERVER_CREDENTIALS](#)

1.11.7 Pagure-Merge-Requests

Neu in Version 4.3.2.

This just adds a thin layer atop [Git](#) using the [Pagure API](#) to allow pushing translation changes as merge requests instead of pushing directly to the repository.

Es besteht keine Notwendigkeit, dies für den Zugriff auf Git-Repositorys zu verwenden, das gewöhnliche [Git](#) funktioniert genauso, der einzige Unterschied ist, wie das Pushen zu einem Repository gehandhabt wird. Mit [Git](#) werden Änderungen direkt in das Repository gepusht, während [Pagure-Merge-Requests](#) einen Merge Request erstellt.

You need to configure API credentials ([PAGURE_CREDENTIALS](#)) in the Weblate settings to make this work. Once configured, you will see a [Pagure](#) option when selecting [Versionsverwaltung](#).

Siehe auch:

Pushen von Änderungen aus Weblate, [PAGURE_CREDENTIALS](#)

1.11.8 Gerrit

Neu in Version 2.2.

Adds a thin layer atop [Git](#) using the [git-review](#) tool to allow pushing translation changes as Gerrit review requests, instead of pushing them directly to the repository.

Die Gerrit-Dokumentation enthält Einzelheiten zur Konfiguration, die für die Einrichtung solcher Repositorys erforderlich ist.

1.11.9 Mercurial

Neu in Version 2.1.

Mercurial is another VCS you can use directly in Weblate.

Bemerkung: It should work with any Mercurial version, but there are sometimes incompatible changes to the command-line interface which breaks Weblate integration.

Siehe auch:

See [Zugriff auf Repositorys](#) for info on how to access different kinds of repositories.

1.11.10 Subversion

Neu in Version 2.8.

Weblate uses [git-svn](#) to interact with [subversion](#) repositories. It is a Perl script that lets subversion be used by a Git client, enabling users to maintain a full clone of the internal repository and commit locally.

Bemerkung: Weblate tries to detect Subversion repository layout automatically - it supports both direct URLs for branch or repositories with standard layout (branches/, tags/ and trunk/). More info about this is to be found in the [git-svn documentation](#). If your repository does not have a standard layout and you encounter errors, try including the branch name in the repository URL and leaving branch empty.

Geändert in Version 2.19: Before this, only repositories using the standard layout were supported.

Subversion credentials

Weblate expects you to have accepted the certificate up-front (and your credentials if needed). It will look to insert them into the `DATA_DIR` directory. Accept the certificate by using `svn` once with the `$HOME` environment variable set to the `DATA_DIR`:

```
# Use DATA_DIR as configured in Weblate settings.py, it is /app/data in the Docker
HOME=${DATA_DIR}/home svn co https://svn.example.com/example
```

Siehe auch:

`DATA_DIR`

1.11.11 Lokale Dateien

1.11.12 Git

Hinweis: Darunter verwendet es [Git](#). Es erfordert die Installation von Git und ermöglicht es Ihnen, Git nativ mit einem vollständigen Verlauf Ihrer Übersetzungen zu verwenden.

Neu in Version 3.8.

Weblate can also operate without a remote VCS. The initial translations are imported by uploading them. Later you can replace individual files by file upload, or add translation strings directly from Weblate (currently available only for monolingual translations).

In the background Weblate creates a Git repository for you and all changes are tracked in. In case you later decide to use a VCS to store the translations, you already have a repository within Weblate can base your integration on.

1.12 Weblates REST API

Neu in Version 2.6: Die REST API ist seit Weblate 2.6 verfügbar.

Die API ist unter der URL `/api/` erreichbar und basiert auf dem [Django-REST-Framework](#). Sie können sie direkt oder über den [Weblate-Client](#) verwenden.

1.12.1 Authentifizierung und allgemeine Parameter

Die öffentliche Projekt-API ist ohne Authentifizierung verfügbar, allerdings werden unauthentifizierte Anfragen stark gedrosselt (standardmäßig auf 100 Anfragen pro Tag), so dass eine Authentifizierung empfohlen wird. Für die Authentifizierung wird ein Token verwendet, das Sie in Ihrem Profil erhalten können. Verwenden Sie es in der `Authorization`-Kopfzeile:

ANY /

Das allgemeine Anfrageverhalten für die API, die Header, Statuscodes und Parameter hier gelten für alle Endpunkte gleichermaßen.

Abfrageparameter

- **format** – Antwortformat (hat Vorrang vor `Accept`). Mögliche Werte hängen von der Einrichtung des REST-Frameworks ab, standardmäßig werden `json` und `api` unterstützt. Letzteres bietet eine Webbrowser-Schnittstelle für die API.
- **page** – Gibt eine Seite mit paginierten Ergebnissen zurück (verwenden Sie die Felder *next* und *previous* in der Antwort, um die Navigation zu automatisieren).

Anfrage-Kopfzeile

- **Accept** – der Inhaltstyp der Antwort hängt vom `Accept`-Header ab
- **Authorization** – optionales Token zur Authentifizierung als `Authorization: Token YOUR-TOKEN`

Antwort-Kopfzeile

- **Content-Type** – dies hängt vom `Accept`-Header der Anfrage ab
- **Allow** – aufgelistete zulässige HTTP-Methoden für das Objekt

JSON Object-Antwort

- **detail** (*string*) – ausführliche Beschreibung des Ergebnisses (für andere HTTP-Statuscodes als 200 OK)
- **count** (*int*) – gesamte Anzahl der Elemente für Objektlisten
- **next** (*string*) – URL der nächsten Seite für Objektlisten
- **previous** (*string*) – URL der vorherigen Seite für Objektlisten
- **results** (*array*) – Ergebnisse für Objektlisten
- **url** (*string*) – URL für den Zugriff auf diese Ressource über die API
- **web_url** (*string*) – URL für den Zugriff auf diese Ressource mit einem Webbrowser

Statuscodes

- 200 OK – wenn die Anfrage korrekt bearbeitet wurde
- 201 Created – wenn ein neues Objekt erfolgreich erstellt wurde
- 204 No Content – wenn ein Objekt erfolgreich gelöscht wurde
- 400 Bad Request – wenn Formularparameter fehlen
- 403 Forbidden – wenn der Zugriff verweigert wird
- 429 Too Many Requests – wenn die Drosselung in Kraft ist

Authentifizierungstoken

Geändert in Version 4.10: Projektspezifische Token wurden in der Version 4.10 eingeführt.

Jeder Benutzer hat sein persönliches Zugangstoken, das im Benutzerprofil abgerufen werden kann. Neu generierte Benutzertoken haben das Präfix `wlu_`.

Es ist möglich, projektspezifische Token für den API-Zugang nur für ein bestimmtes Projekt zu erstellen. Diese Token können durch das Präfix `wlp_` identifiziert werden.

Beispiele für die Authentifizierung

Beispielanfrage:

```
GET /api/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
Authorization: Token YOUR-TOKEN
```

Beispielantwort:

```
HTTP/1.0 200 OK
Date: Fri, 25 Mar 2016 09:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, HEAD, OPTIONS

{
  "projects": "http://example.com/api/projects/",
  "components": "http://example.com/api/components/",
  "translations": "http://example.com/api/translations/",
  "languages": "http://example.com/api/languages/"
}
```

CURL-Beispiel:

```
curl \
  -H "Authorization: Token TOKEN" \
  https://example.com/api/
```

Beispiele für die Parameterübergabe

Bei der `POST`-Methode können die Parameter entweder als Formularübermittlung (*application/x-www-form-urlencoded*) oder als JSON (*application/json*) angegeben werden.

Formular-Anfrage-Beispiel:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Token TOKEN

operation=pull
```

JSON-Anfrage-Beispiel:


```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"operation": "pull"}
```

CURL-Beispiel:

```
curl \
  -d operation=pull \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

CURL-JSON-Beispiel:

```
curl \
  --data-binary '{"operation": "pull"}' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

API-Ratenbegrenzung

Die Anzahl der API-Anfragen ist begrenzt; die Standardkonfiguration beschränkt sie auf 100 Anfragen pro Tag für anonyme Benutzer und 5000 Anfragen pro Stunde für authentifizierte Benutzer.

Die Ratenbegrenzung kann in `settings.py` eingestellt werden; siehe [Drosselung in Django REST-Framework Dokumentation](#) für weitere Details zur Konfiguration.

Im Docker-Container kann dies mit `WEBLATE_API_RATELIMIT_ANON` und `WEBLATE_API_RATELIMIT_USER` konfiguriert werden.

Der Status der Ratenbegrenzung wird in den folgenden Kopfzeilen gemeldet:

X-RateLimit-Limit	Limit der Anzahl der auszuführenden Anfragen
X-RateLimit-Remaining	Verbleibendes Limit für Anfragen
X-RateLimit-Reset	Anzahl der Sekunden bis zum Zurücksetzen des Ratenbegrenzungsfensters

Geändert in Version 4.1: Statuskopfzeilen für die Ratenbegrenzung hinzugefügt.

Siehe auch:

[Ratenbegrenzung](#), [Ratenbegrenzung](#), `WEBLATE_API_RATELIMIT_ANON`, `WEBLATE_API_RATELIMIT_USER`

1.12.2 API-Einstiegspunkt**GET /api/**

Der API-Root-Einstiegspunkt.

Beispielanfrage:

```
GET /api/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
Authorization: Token YOUR-TOKEN
```

Beispielantwort:

```
HTTP/1.0 200 OK
Date: Fri, 25 Mar 2016 09:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, HEAD, OPTIONS

{
  "projects": "http://example.com/api/projects/",
  "components": "http://example.com/api/components/",
  "translations": "http://example.com/api/translations/",
  "languages": "http://example.com/api/languages/"
}
```

1.12.3 Benutzer

Neu in Version 4.0.

GET /api/users/

Gibt eine Liste der Benutzer zurück, wenn Sie die Berechtigung haben, Benutzer zu verwalten. Wenn nicht, sehen Sie nur Ihre eigenen Details.

Siehe auch:

Die Objektattribute von Benutzern sind unter `GET /api/users/(str:username)/` dokumentiert.

POST /api/users/

Erstellt einen neuen Benutzer.

Parameter

- **username** (*string*) – Benutzername
- **full_name** (*string*) – Vollständiger Name des Benutzers
- **email** (*string*) – E-Mail-Adresse des Benutzers
- **is_superuser** (*boolean*) – Ist der Benutzer Superuser? (Optional)
- **is_active** (*boolean*) – Ist der Benutzer aktiv? (Optional)
- **is_bot** (*boolean*) – Ist der Benutzer ein Bot? (Optional) (wird für projektspezifische Token verwendet)

GET /api/users/(str: username) /

Gibt Informationen über Benutzer zurück.

Parameter

- **username** (*string*) – Name des Benutzers

JSON Object-Antwort

- **username** (*string*) – Benutzername eines Benutzers
- **full_name** (*string*) – Vollständiger Name eines Benutzers
- **email** (*string*) – E-Mail-Adresse eines Benutzers
- **is_superuser** (*boolean*) – Ob der Benutzer ein Superuser ist
- **is_active** (*boolean*) – Ob der Benutzer aktiv ist

- **is_bot** (*boolean*) – Ob der Benutzer ein Bot ist (wird für projektspezifische Token verwendet)
- **date_joined** (*string*) – Datum der Erstellung des Benutzers
- **groups** (*array*) – Link zu den zugeordneten Gruppen; siehe `GET /api/groups/ (int:id)/`

JSON-Daten-Beispiel:

```
{
  "email": "user@example.com",
  "full_name": "Example User",
  "username": "exampleusername",
  "groups": [
    "http://example.com/api/groups/2/",
    "http://example.com/api/groups/3/"
  ],
  "is_superuser": true,
  "is_active": true,
  "is_bot": false,
  "date_joined": "2020-03-29T18:42:42.617681Z",
  "url": "http://example.com/api/users/exampleusername/",
  "statistics_url": "http://example.com/api/users/exampleusername/statistics/"
}
```

PUT /api/users/ (str: username) /

Ändert die Benutzerparameter.

Parameter

- **username** (*string*) – Name des Benutzers

JSON Object-Antwort

- **username** (*string*) – Benutzername eines Benutzers
- **full_name** (*string*) – Vollständiger Name eines Benutzers
- **email** (*string*) – E-Mail-Adresse eines Benutzers
- **is_superuser** (*boolean*) – Ob der Benutzer ein Superuser ist
- **is_active** (*boolean*) – Ob der Benutzer aktiv ist
- **is_bot** (*boolean*) – Ob der Benutzer ein Bot ist (wird für projektspezifische Token verwendet)
- **date_joined** (*string*) – Datum der Erstellung des Benutzers

PATCH /api/users/ (str: username) /

Ändert die Benutzerparameter.

Parameter

- **username** (*string*) – Name des Benutzers

JSON Object-Antwort

- **username** (*string*) – Benutzername eines Benutzers
- **full_name** (*string*) – Vollständiger Name eines Benutzers
- **email** (*string*) – E-Mail-Adresse eines Benutzers
- **is_superuser** (*boolean*) – Ob der Benutzer ein Superuser ist
- **is_active** (*boolean*) – Ob der Benutzer aktiv ist

- **is_bot** (*boolean*) – Ob der Benutzer ein Bot ist (wird für projektspezifische Token verwendet)
- **date_joined** (*string*) – Datum der Erstellung des Benutzers

DELETE /api/users/ (str: username) /

Löscht alle Benutzerinformationen und markiert den Benutzer als inaktiv.

Parameter

- **username** (*string*) – Name des Benutzers

POST /api/users/ (str: username) /groups/

Ordnet Gruppen einem Benutzer zu.

Parameter

- **username** (*string*) – Name des Benutzers

Formularparameter

- **string group_id** – Die eindeutige Gruppen-ID

DELETE /api/users/ (str: username) /groups/

Neu in Version 4.13.1.

Entfernt Benutzer aus einer Gruppe.

Parameter

- **username** (*string*) – Name des Benutzers

Formularparameter

- **string group_id** – Die eindeutige Gruppen-ID

GET /api/users/ (str: username) /statistics/

Listet die Statistik eines Benutzers auf.

Parameter

- **username** (*string*) – Name des Benutzers

JSON Object-Antwort

- **translated** (*int*) – Anzahl der Übersetzungen des Benutzers
- **suggested** (*int*) – Anzahl der Vorschläge des Benutzers
- **uploaded** (*int*) – Anzahl der Uploads des Benutzers
- **commented** (*int*) – Anzahl der Kommentare des Benutzers
- **languages** (*int*) – Anzahl der Sprachen, die der Benutzer übersetzen kann

GET /api/users/ (str: username) /notifications/

Listet die Abonnements eines Benutzers auf.

Parameter

- **username** (*string*) – Name des Benutzers

POST /api/users/ (str: username) /notifications/

Ordnet Abonnements einem Benutzer zu.

Parameter

- **username** (*string*) – Name des Benutzers

JSON Object-Anfrage

- **notification** (*string*) – Name der registrierten Benachrichtigung

- **scope** (*int*) – Umfang der Benachrichtigung aus den verfügbaren Auswahlmöglichkeiten
- **frequency** (*int*) – Auswahl der Häufigkeit von Benachrichtigungen

GET /api/users/ (**str**: *username*) /notifications/
int: *subscription_id*/

Ruft das einem Benutzer zugeordnete Abonnement ab.

Parameter

- **username** (*string*) – Name des Benutzers
- **subscription_id** (*int*) – ID der registrierten Benachrichtigung

PUT /api/users/ (**str**: *username*) /notifications/
int: *subscription_id*/

Bearbeitet das einem Benutzer zugeordnete Abonnement.

Parameter

- **username** (*string*) – Name des Benutzers
- **subscription_id** (*int*) – ID der registrierten Benachrichtigung

JSON Object-Anfrage

- **notification** (*string*) – Name der registrierten Benachrichtigung
- **scope** (*int*) – Umfang der Benachrichtigung aus den verfügbaren Auswahlmöglichkeiten
- **frequency** (*int*) – Auswahl der Häufigkeit von Benachrichtigungen

PATCH /api/users/ (**str**: *username*) /notifications/
int: *subscription_id*/

Bearbeitet das einem Benutzer zugeordnete Abonnement.

Parameter

- **username** (*string*) – Name des Benutzers
- **subscription_id** (*int*) – ID der registrierten Benachrichtigung

JSON Object-Anfrage

- **notification** (*string*) – Name der registrierten Benachrichtigung
- **scope** (*int*) – Umfang der Benachrichtigung aus den verfügbaren Auswahlmöglichkeiten
- **frequency** (*int*) – Auswahl der Häufigkeit von Benachrichtigungen

DELETE /api/users/ (**str**: *username*) /notifications/
int: *subscription_id*/

Löscht das einem Benutzer zugeordnete Abonnement.

Parameter

- **username** (*string*) – Name des Benutzers
- **subscription_id** – Name der registrierten Benachrichtigung
- **subscription_id** – Int

1.12.4 Gruppen

Neu in Version 4.0.

GET /api/groups/

Gibt eine Liste der Gruppen zurück, wenn Sie die Berechtigung haben, Gruppen zu verwalten. Wenn nicht, sehen Sie nur die Gruppen, denen der Benutzer angehört.

Siehe auch:

Die Objektattribute von Gruppen sind unter [GET /api/groups/\(int:id\)/](#) dokumentiert.

POST /api/groups/

Erstellt eine neue Gruppe.

Parameter

- **name** (*string*) – Gruppenname
- **project_selection** (*int*) – Gruppe der Projektauswahl aus vorgegebenen Optionen
- **language_selection** (*int*) – Gruppe von Sprachen aus vorgegebenen Optionen
- **defining_project** (*str*) – Link zum definierenden Projekt, verwendet für die *Verwaltung der projektspezifischen Zugriffssteuerung*; siehe [GET /api/projects/\(string:project\)/](#)

GET /api/groups/(int: id)/

Gibt Informationen über die Gruppe zurück.

Parameter

- **id** (*int*) – ID der Gruppe

JSON Object-Antwort

- **name** (*string*) – Name einer Gruppe
- **project_selection** (*int*) – Ganzzahl, entsprechend einer Gruppe von Projekten
- **language_selection** (*int*) – Ganzzahl, entsprechend einer Gruppe von Sprachen
- **roles** (*array*) – Link zu den zugeordneten Rollen; siehe [GET /api/roles/\(int:id\)/](#)
- **projects** (*array*) – Link zu den zugeordneten Projekten; siehe [GET /api/projects/\(string:project\)/](#)
- **components** (*array*) – Link zu den zugeordneten Komponenten; siehe [GET /api/components/\(string:project\)/\(string:component\)/](#)
- **componentlists** (*array*) – Link zur zugeordneten Komponentenliste; siehe [GET /api/component-lists/\(str:slug\)/](#)
- **defining_project** (*str*) – Link zum definierenden Projekt, verwendet für die *Verwaltung der projektspezifischen Zugriffssteuerung*; siehe [GET /api/projects/\(string:project\)/](#)

JSON-Daten-Beispiel:

```
{
  "name": "Guests",
  "defining_project": null,
  "project_selection": 3,
  "language_selection": 1,
  "url": "http://example.com/api/groups/1/",
  "roles": [
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    "http://example.com/api/roles/1/",
    "http://example.com/api/roles/2/"
  ],
  "languages": [
    "http://example.com/api/languages/en/",
    "http://example.com/api/languages/cs/"
  ],
  "projects": [
    "http://example.com/api/projects/demo1/",
    "http://example.com/api/projects/demo/"
  ],
  "componentlist": "http://example.com/api/component-lists/new/",
  "components": [
    "http://example.com/api/components/demo/weblate/"
  ]
}

```

PUT /api/groups/ (int: id) /

Ändert die Gruppenparameter.

Parameter

- **id** (*int*) – ID der Gruppe

JSON Object-Antwort

- **name** (*string*) – Name einer Gruppe
- **project_selection** (*int*) – Ganzzahl, entsprechend einer Gruppe von Projekten
- **language_selection** (*int*) – Ganzzahl, entsprechend einer Gruppe von Sprachen

PATCH /api/groups/ (int: id) /

Ändert die Gruppenparameter.

Parameter

- **id** (*int*) – ID der Gruppe

JSON Object-Antwort

- **name** (*string*) – Name einer Gruppe
- **project_selection** (*int*) – Ganzzahl, entsprechend einer Gruppe von Projekten
- **language_selection** (*int*) – Ganzzahl, entsprechend einer Gruppe von Sprachen

DELETE /api/groups/ (int: id) /

Löscht die Gruppe.

Parameter

- **id** (*int*) – ID der Gruppe

POST /api/groups/ (int: id) /roles/

Ordnet Rollen einer Gruppe zu.

Parameter

- **id** (*int*) – ID der Gruppe

Formularparameter

- **string role_id** – Die eindeutige Rollen-ID

POST /api/groups/ (int: id) /components/

Ordnet Komponenten einer Gruppe zu.

Parameter

- **id** (*int*) – ID der Gruppe

Formularparameter

- **string component_id** – Die eindeutige Komponenten-ID

DELETE /api/groups/(int: id)/components/

int: *component_id*

Löscht die Komponente aus einer Gruppe.

Parameter

- **id** (*int*) – ID der Gruppe
- **component_id** (*int*) – Die eindeutige Komponenten-ID

POST /api/groups/(int: id)/projects/

Ordnet Projekte einer Gruppe zu.

Parameter

- **id** (*int*) – ID der Gruppe

Formularparameter

- **string project_id** – Die eindeutige Projekt-ID

DELETE /api/groups/(int: id)/projects/

int: *project_id*

Löscht das Projekt aus einer Gruppe.

Parameter

- **id** (*int*) – ID der Gruppe
- **project_id** (*int*) – Die eindeutige Projekt-ID

POST /api/groups/(int: id)/languages/

Ordnet Sprachen einer Gruppe zu.

Parameter

- **id** (*int*) – ID der Gruppe

Formularparameter

- **string language_code** – Der eindeutige Sprachcode

DELETE /api/groups/(int: id)/languages/

string: *language_code*

Löscht die Sprache aus einer Gruppe.

Parameter

- **id** (*int*) – ID der Gruppe
- **language_code** (*string*) – Der eindeutige Sprachcode

POST /api/groups/(int: id)/componentlists/

Ordnet Komponentenlisten einer Gruppe zu.

Parameter

- **id** (*int*) – ID der Gruppe

Formularparameter

- **string component_list_id** – Die eindeutige Komponentenlisten-ID

DELETE /api/groups/(int: id)/componentlists/

int: *component_list_id*

Löscht die Komponentenliste aus einer Gruppe.

Parameter

- **id** (*int*) – ID der Gruppe
- **component_list_id** (*int*) – Die eindeutige Komponentenlisten-ID

1.12.5 Rollen

GET /api/roles/

Gibt eine Liste aller mit dem Benutzer zugeordneten Rollen zurück. Wenn der Benutzer ein Superuser ist, wird eine Liste aller vorhandenen Rollen zurückgegeben.

Siehe auch:

Die Objektattribute von Rollen sind unter `GET /api/roles/(int:id)/` dokumentiert.

POST /api/roles/

Erstellt eine neue Rolle.

Parameter

- **name** (*string*) – Rollenname
- **permissions** (*array*) – Liste der Codenamen der Berechtigungen

GET /api/roles/(int: id) /

Gibt Informationen über eine Rolle zurück.

Parameter

- **id** (*int*) – ID der Rolle

JSON Object-Antwort

- **name** (*string*) – Rollenname
- **permissions** (*array*) – Liste der Codenamen der Berechtigungen

JSON-Daten-Beispiel:

```
{
  "name": "Access repository",
  "permissions": [
    "vcs.access",
    "vcs.view"
  ],
  "url": "http://example.com/api/roles/1/",
}
```

PUT /api/roles/(int: id) /

Ändert die Rollenparameter.

Parameter

- **id** (*int*) – ID der Rolle

JSON Object-Antwort

- **name** (*string*) – Rollenname
- **permissions** (*array*) – Liste der Codenamen der Berechtigungen

PATCH `/api/roles/(int: id) /`

Ändert die Rollenparameter.

Parameter

- **id** (*int*) – ID der Rolle

JSON Object-Antwort

- **name** (*string*) – Rollenname
- **permissions** (*array*) – Liste der Codenamen der Berechtigungen

DELETE `/api/roles/(int: id) /`

Löscht die Rolle.

Parameter

- **id** (*int*) – ID der Rolle

1.12.6 Sprachen

GET `/api/languages/`

Gibt eine Liste aller Sprachen zurück.

Siehe auch:

Die Objektattribute von Sprachen sind unter `GET /api/languages/(string: language) /` dokumentiert.

POST `/api/languages/`

Erstellt eine neue Sprache.

Parameter

- **code** (*string*) – Sprachename
- **name** (*string*) – Sprachename
- **direction** (*string*) – Leserichtung
- **population** (*int*) – Anzahl der Sprecher
- **plural** (*object*) – Pluralformen und -anzahl der Sprache

GET `/api/languages/(string: language) /`

Gibt Informationen über eine Sprache zurück.

Parameter

- **language** (*string*) – Sprachcode

JSON Object-Antwort

- **code** (*string*) – Sprachcode
- **direction** (*string*) – Leserichtung
- **plural** (*object*) – Object der Pluralinformationen der Sprache
- **aliases** (*array*) – Array der Sprachaliasnamen

JSON Object-Anfrage

- **population** (*int*) – Anzahl der Sprecher

JSON-Daten-Beispiel:

```
{
  "code": "en",
  "direction": "ltr",
  "name": "English",
  "population": 159034349015,
  "plural": {
    "id": 75,
    "source": 0,
    "number": 2,
    "formula": "n != 1",
    "type": 1
  },
  "aliases": [
    "english",
    "en_en",
    "base",
    "source",
    "eng"
  ],
  "url": "http://example.com/api/languages/en/",
  "web_url": "http://example.com/languages/en/",
  "statistics_url": "http://example.com/api/languages/en/statistics/"
}
```

PUT `/api/languages/(string: language)/`

Ändert die Sprachparameter.

Parameter

- **language** (*string*) – Sprachcode

JSON Object-Anfrage

- **name** (*string*) – Sprachname
- **direction** (*string*) – Leserichtung
- **population** (*int*) – Anzahl der Sprecher
- **plural** (*object*) – Details zu Pluralformen der Sprache

PATCH `/api/languages/(string: language)/`

Ändert die Sprachparameter.

Parameter

- **language** (*string*) – Sprachcode

JSON Object-Anfrage

- **name** (*string*) – Sprachname
- **direction** (*string*) – Leserichtung
- **population** (*int*) – Anzahl der Sprecher
- **plural** (*object*) – Details zu Pluralformen der Sprache

DELETE `/api/languages/(string: language)/`

Löscht die Sprache.

Parameter

- **language** (*string*) – Sprachcode

GET `/api/languages/(string: language)/statistics/`

Liefert Statistiken für eine Sprache.

Parameter

- **language** (*string*) – Sprachcode

JSON Object-Antwort

- **total** (*int*) – Gesamtzahl der Zeichenketten
- **total_words** (*int*) – Gesamtzahl der Wörter
- **last_change** (*timestamp*) – Letzte Änderungen in der Sprache
- **recent_changes** (*int*) – Gesamtzahl der Änderungen
- **translated** (*int*) – Anzahl übersetzter Zeichenketten
- **translated_percent** (*float*) – Prozentsatz der übersetzten Zeichenketten
- **translated_words** (*int*) – Anzahl der übersetzten Wörter
- **translated_words_percent** (*int*) – Prozentsatz der übersetzten Wörter
- **translated_chars** (*int*) – Anzahl der übersetzten Zeichen
- **translated_chars_percent** (*int*) – Prozentsatz der übersetzten Zeichen
- **total_chars** (*int*) – Gesamtzahl der Zeichen
- **fuzzy** (*int*) – Anzahl der fragwürdigen (zur Bearbeitung markierten) Zeichenketten
- **fuzzy_percent** (*int*) – Prozentsatz der fragwürdigen (zur Bearbeitung markierten) Zeichenketten
- **failing** (*int*) – Anzahl der fehlerhaften Zeichenketten
- **failing** – Prozentsatz der fehlerhaften Zeichenketten

1.12.7 Projekte

GET /api/projects/

Gibt eine Liste aller Projekte zurück.

Siehe auch:

Die Objektattribute von Projekten sind unter `GET /api/projects/(string:project)/` dokumentiert.

POST /api/projects/

Neu in Version 3.9.

Erstellt ein neues Projekt.

Parameter

- **name** (*string*) – Projektname
- **slug** (*string*) – Projekt-Slug
- **web** (*string*) – Projektseite

GET /api/projects/(string: project) /

Gibt Informationen über ein Projekt zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

JSON Object-Antwort

- **name** (*string*) – Projektname
- **slug** (*string*) – Projekt-Slug

- **web** (*string*) – Projektseite
- **components_list_url** (*string*) – URL zur Komponentenliste; siehe `GET /api/projects/(string:project)/components/`
- **repository_url** (*string*) – URL zum Repository-Status; siehe `GET /api/projects/(string:project)/repository/`
- **changes_list_url** (*string*) – URL zur Änderungsliste; siehe `GET /api/projects/(string:project)/changes/`
- **translation_review** (*boolean*) – *Überprüfung aktivieren*
- **source_review** (*boolean*) – *Quellenüberprüfung aktivieren*
- **set_language_team** (*boolean*) – *Kopfzeile „Language-Team“ setzen*
- **enable_hooks** (*boolean*) – *Hooks aktivieren*
- **instructions** (*string*) – *Übersetzungsanweisungen*
- **language_aliases** (*string*) – *Sprachaliasnamen*

JSON-Daten-Beispiel:

```
{
  "name": "Hello",
  "slug": "hello",
  "url": "http://example.com/api/projects/hello/",
  "web": "https://weblate.org/",
  "web_url": "http://example.com/projects/hello/"
}
```

PATCH /api/projects/(string: project) /

Neu in Version 4.3.

Bearbeitet ein Projekt durch eine **PATCH**-Anfrage.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

PUT /api/projects/(string: project) /

Neu in Version 4.3.

Bearbeitet ein Projekt durch eine **PUT**-Anfrage.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

DELETE /api/projects/(string: project) /

Neu in Version 3.9.

Löscht ein Projekt.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

GET /api/projects/(string: project) /changes/

Gibt eine Liste der Projektänderungen zurück. Dies ist im Wesentlichen ein projektspezifisches `GET /api/changes/`, das dieselben Parameter akzeptiert.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

JSON Object-Antwort

- **results** (*array*) – Array von Komponentenobjekten; siehe `GET /api/changes/(int:id)/`

GET /api/projects/(string: project)/repository/

Gibt Informationen über den Status des VCS-Repositorys zurück. Dieser Endpunkt enthält nur eine Gesamtzusammenfassung für alle Repositorys für das Projekt. Um einen detaillierteren Status zu erhalten, verwenden Sie `GET /api/components/(string:project)/(string:component)/repository/`.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

JSON Object-Antwort

- **needs_commit** (*boolean*) – Ob es irgendwelche ausstehenden Änderungen zu committen gibt
- **needs_merge** (*boolean*) – Ob es irgendwelche Upstream-Änderungen zum Zusammenführen gibt
- **needs_push** (*boolean*) – Ob es irgendwelche lokale Änderungen zum Pushen gibt

JSON-Daten-Beispiel:

```
{
  "needs_commit": true,
  "needs_merge": false,
  "needs_push": true
}
```

POST /api/projects/(string: project)/repository/

Führt eine bestimmte Operation im VCS-Repository durch.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

JSON Object-Anfrage

- **operation** (*string*) – Auszuführende Operation: eine von push, pull, commit, reset, cleanup, file-sync

JSON Object-Antwort

- **result** (*boolean*) – Ergebnis der Operation

CURL-Beispiel:

```
curl \
  -d operation=pull \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/repository/
```

JSON-Anfrage-Beispiel:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"operation": "pull"}
```

JSON-Antwort-Beispiel:

```

HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"result":true}

```

GET `/api/projects/(string: project)/components/`

Gibt eine Liste der Übersetzungskomponenten im angegebenen Projekt zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

JSON Object-Antwort

- **results** (*array*) – Array von Komponentenobjekten; siehe `GET /api/components/(string:project)/(string:component)/`

POST `/api/projects/(string: project)/components/`

Neu in Version 3.9.

Geändert in Version 4.3: Die Parameter `zipfile` und `docfile` werden jetzt für VCS-freie Komponenten akzeptiert, siehe [Lokale Dateien](#).

Geändert in Version 4.6: Die geklonten Repositorys werden nun automatisch innerhalb eines Projekts mit [Weblate-interne URLs](#) geteilt. Verwenden Sie `disable_autoshare` um dies zu deaktivieren.

Erstellt Übersetzungskomponenten im angegebenen Projekt.

Hinweis: Verwenden Sie [Weblate-interne URLs](#), wenn Sie mehrere Komponenten aus einem einzigen VCS-Repository erstellen.

Bemerkung: Der größte Teil der Komponentenerstellung erfolgt im Hintergrund. Prüfen Sie das Attribut `task_url` der erstellten Komponente und verfolgen Sie dort den Fortschritt.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

Formularparameter

- **file zipfile** – ZIP-Datei zum Hochladen in Weblate für die Initialisierung der Übersetzungen
- **file docfile** – Zu übersetzendes Dokument
- **boolean disable_autoshare** – Deaktiviert das automatische Teilen des Repositorys über [Weblate-interne URLs](#).

JSON Object-Anfrage

- **object** – Komponentenparameter, siehe `GET /api/components/(string:project)/(string:component)/`

JSON Object-Antwort

- **result** (*object*) – Erstelltes Komponentenobjekt; siehe `GET /api/components/(string:project)/(string:component)/`

JSON kann nicht verwendet werden, wenn die Dateien mit den Parametern `zipfile` und `docfile` hochgeladen werden. Die Daten müssen als *multipart/form-data* hochgeladen werden.

CURL-Formular-Anfrage-Beispiel:

```
curl \
  --form docfile=@strings.html \
  --form name=Weblate \
  --form slug=weblate \
  --form file_format=html \
  --form new_lang=add \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

CURL-JSON-Anfrage-Beispiel:

```
curl \
  --data-binary '{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "https://github.com/WeblateOrg/hello.git",
    "template": "",
    "new_base": "po/hello.pot",
    "vcs": "git"
  }' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

JSON-Anfrage zur Erstellung einer neuen Komponente von Git:

```
POST /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "https://github.com/WeblateOrg/hello.git",
  "template": "",
  "new_base": "po/hello.pot",
  "vcs": "git"
}
```

JSON-Anfrage zum Erstellen einer neuen Komponente aus einer anderen:

```
POST /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
{
  "file_format": "po",
  "filemask": "po/*.po",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "weblate://weblate/hello",
  "template": "",
  "new_base": "po/hello.pot",
  "vcs": "git"
}
```

JSON-Antwort-Beispiel:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "population": 159034349015,
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}
```

GET /api/projects/(string: *project*)/languages/

Gibt paginierte Statistiken für alle Sprachen innerhalb eines Projekts zurück.

Neu in Version 3.8.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

JSON Object-Antwort

- **results** (*array*) – Array von Übersetzungsstatistikobjekten
- **language** (*string*) – Sprachename
- **code** (*string*) – Sprachcode
- **total** (*int*) – Gesamtzahl der Zeichenketten
- **translated** (*int*) – Anzahl übersetzter Zeichenketten
- **translated_percent** (*float*) – Prozentsatz der übersetzten Zeichenketten
- **total_words** (*int*) – Gesamtzahl der Wörter
- **translated_words** (*int*) – Anzahl der übersetzten Wörter
- **words_percent** (*float*) – Prozentsatz der übersetzten Wörter

GET /api/projects/(string: project)/statistics/

Gibt Statistiken für ein Projekt zurück.

Neu in Version 3.8.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

JSON Object-Antwort

- **total** (*int*) – Gesamtzahl der Zeichenketten
- **translated** (*int*) – Anzahl übersetzter Zeichenketten
- **translated_percent** (*float*) – Prozentsatz der übersetzten Zeichenketten
- **total_words** (*int*) – Gesamtzahl der Wörter
- **translated_words** (*int*) – Anzahl der übersetzten Wörter
- **words_percent** (*float*) – Prozentsatz der übersetzten Wörter

1.12.8 Komponenten

Hinweis: Verwenden Sie `POST /api/projects/(string:project)/components/` um neue Komponenten zu erstellen.

GET /api/components/

Gibt eine Liste der Übersetzungskomponenten zurück.

Siehe auch:

Die Objektattribute von Komponenten sind unter `GET /api/components/(string:project)/(string:component)/` dokumentiert.

GET /api/components/(string: project) /
string: component/

Gibt Informationen über die Übersetzungskomponente zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **project** (*object*) – Das Übersetzungsprojekt; siehe `GET /api/projects/(string:project)/`
- **name** (*string*) – *Name der Komponente*
- **slug** (*string*) – *Komponenten-Slug*
- **vcs** (*string*) – *Versionsverwaltung*
- **repo** (*string*) – *Quellcode-Repository*
- **git_export** (*string*) – *Exportierte Paketquelladresse*
- **branch** (*string*) – *Repository-Zweig*
- **push_branch** (*string*) – *Push Branch*
- **filemask** (*string*) – *Dateimaske*
- **template** (*string*) – *Einsprachige Basis-Sprachdatei*
- **edit_template** (*string*) – *Basisdatei bearbeiten*
- **intermediate** (*string*) – *Zwischensprachedatei*
- **new_base** (*string*) – *Vorlage für neue Übersetzungen*
- **file_format** (*string*) – *Dateiformat*
- **license** (*string*) – *Lizenzierung der Übersetzung*
- **agreement** (*string*) – *Mitwirkungsvereinbarung*
- **new_lang** (*string*) – *Neue Übersetzung hinzufügen*
- **language_code_style** (*string*) – *Stil des Sprachcodes*
- **source_language** (*object*) – Ausgangsspracheobjekt; siehe `GET /api/languages/(string:language)/`
- **push** (*string*) – *Push-URL für Repository*
- **check_flags** (*string*) – *Übersetzungsmarkierungen*
- **priority** (*string*) – *Priorität*
- **enforced_checks** (*string*) – *Erzwungene Qualitätsprüfungen*
- **restricted** (*string*) – *Eingeschränkter Zugriff*
- **repoweb** (*string*) – *Paketquellnavigator*
- **report_source_bugs** (*string*) – *Adresse für Fehlerberichte bei Ausgangszeichenkett*
- **merge_style** (*string*) – *Git-Strategie*
- **commit_message** (*string*) – *Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten*
- **add_message** (*string*) – *Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten*
- **delete_message** (*string*) – *Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten*
- **merge_message** (*string*) – *Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten*
- **addon_message** (*string*) – *Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten*
- **pull_message** (*string*) – *Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten*

- **allow_translation_propagation** (*string*) – *Verbreitung von Übersetzungen erlauben*
- **enable_suggestions** (*string*) – *Vorschläge aktivieren*
- **suggestion_voting** (*string*) – *Abstimmen über Vorschläge*
- **suggestion_autoaccept** (*string*) – *Vorschläge automatisch annehmen*
- **push_on_commit** (*string*) – *Bei Commit gleichzeitig Pushen*
- **commit_pending_age** (*string*) – *Alter der Änderungen, das erreicht sein muss, bevor ein Commit erfolgt*
- **auto_lock_error** (*string*) – *Sperre bei Fehler*
- **language_regex** (*string*) – *Sprachen-Filter*
- **variant_regex** (*string*) – *Varianten regulärer Ausdruck*
- **repository_url** (*string*) – URL zum Repository-Status; siehe `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – URL zur Übersetzungsliste; siehe `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – URL zum Sperrstatus; siehe `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – URL zur Änderungsliste; siehe `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – URL zur Hintergrundaufgabe (falls vorhanden); siehe `GET /api/tasks/(str:uuid)/`

JSON-Daten-Beispiel:

```
{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "population": 159034349015,
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "source_language": {
    "code": "en",
    "direction": "ltr",
    "population": 159034349015,
    "name": "English",
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    "url": "http://example.com/api/languages/en/",
    "web_url": "http://example.com/languages/en/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}

```

PATCH /api/components/ (string: project) /
string: component/

Bearbeitet eine Komponente durch eine **PATCH**-Anfrage.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **source_language** (*string*) – Sprachcode der Projekt-Ausgangssprache (optional)

JSON Object-Anfrage

- **name** (*string*) – Name der Komponente
- **slug** (*string*) – Slug der Komponente
- **repo** (*string*) – URL des VCS-Repository

CURL-Beispiel:

```

curl \
  --data-binary '{"name": "new name"}' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  PATCH http://example.com/api/projects/hello/components/

```

JSON-Anfrage-Beispiel:

```

PATCH /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{
  "name": "new name"
}

```

JSON-Antwort-Beispiel:

```

HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

```

(Fortsetzung auf der nächsten Seite)

```
{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "new name",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "population": 159034349015,
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}
```

PUT `/api/components/(string: project) /`
string: `component/`

Bearbeitet eine Komponente durch eine PUT-Anfrage.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Anfrage

- **branch** (*string*) – Branch des VCS-Repository
- **file_format** (*string*) – Dateiformat der Übersetzungen
- **filemask** (*string*) – Maske der Übersetzungsdateien im Repository
- **name** (*string*) – Name der Komponente
- **slug** (*string*) – Slug der Komponente
- **repo** (*string*) – URL des VCS-Repository
- **template** (*string*) – Basisdatei für einsprachige Übersetzungen
- **new_base** (*string*) – Basisdatei für das Hinzufügen neuer Übersetzungen
- **vcs** (*string*) – Versionsverwaltung

DELETE `/api/components/(string: project) /`
string: `component/`

Neu in Version 3.9.

Löscht eine Komponente.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

GET /api/components/ (**string:** *project*) /
string: *component/changes/*

Gibt eine Liste der Komponentenänderungen zurück. Dies ist im Wesentlichen ein komponentenspezifisches [GET /api/changes/](#), das dieselben Parameter akzeptiert.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **results** (*array*) – Array von Komponentenobjekten; siehe [GET /api/changes/ \(int:id\)/](#)

GET /api/components/ (**string:** *project*) /
string: *component/file/*

Neu in Version 4.9.

Lädt alle verfügbaren Übersetzungen, die der Komponente zugeordnet sind, als Archivdatei im gewünschten Format herunter.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

Abfrageparameter

- **format** (*string*) – Das zu verwendende Archivformat; wenn nicht angegeben, wird `zip` als Standardwert verwendet; Unterstützte Formate: `zip`

GET /api/components/ (**string:** *project*) /
string: *component/screenshots/*

Gibt eine Liste der Komponenten-Bildschirmfotos zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **results** (*array*) – Reihe von Komponenten-Bildschirmfotos; siehe [GET /api/screenshots/ \(int:id\)/](#)

GET /api/components/ (**string:** *project*) /
string: *component/lock/*

Gibt den Status der Komponentensperre zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **locked** (*boolean*) – Ob die Komponente für Aktualisierungen gesperrt ist

JSON-Daten-Beispiel:

```
{
  "locked": false
}
```

POST `/api/components/(string: project) /`
`string: component/lock/`

Setzt den Status der Komponentensperre.

Die Antwort ist dieselbe wie bei `GET /api/components/(string:project) /`
`(string:component)/lock/`.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Anfrage

- **lock** – Boolesche Angabe, ob gesperrt werden soll oder nicht.

CURL-Beispiel:

```
curl \
  -d lock=true \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

JSON-Anfrage-Beispiel:

```
POST /api/components/hello/weblate/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"lock": true}
```

JSON-Antwort-Beispiel:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"locked": true}
```

GET `/api/components/(string: project) /`
`string: component/repository/`

Gibt Informationen über den Status des VCS-Repositorys zurück.

Die Antwort ist dieselbe wie bei `GET /api/projects/(string:project)/repository/`.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **needs_commit** (*boolean*) – Ob es irgendwelche ausstehenden Änderungen zu committen gibt
- **needs_merge** (*boolean*) – Ob es irgendwelche Upstream-Änderungen zum Zusammenführen gibt
- **needs_push** (*boolean*) – Ob es irgendwelche lokale Änderungen zum Pushen gibt
- **remote_commit** (*string*) – Information des Remote-Commit
- **status** (*string*) – Repository-Status des VCS, wie von VCS gemeldet
- **merge_failure** – Text des Merge-Konflikts, oder Null, wenn es keinen gibt

POST /api/components/ (*string: project*) /
string: component/repository/

Führt die angegebene Operation an einem VCS-Repository durch.

Siehe *POST /api/projects/(string:project)/repository/* für die Dokumentation.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Anfrage

- **operation** (*string*) – Auszuführende Operation: eine von push, pull, commit, reset, cleanup

JSON Object-Antwort

- **result** (*boolean*) – Ergebnis der Operation

CURL-Beispiel:

```
curl \
  -d operation=pull \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/components/hello/weblate/repository/
```

JSON-Anfrage-Beispiel:

```
POST /api/components/hello/weblate/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20

{"operation":"pull"}
```

JSON-Antwort-Beispiel:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"result":true}
```

GET /api/components/ (*string: project*) /
string: component/monolingual_base/

Lädt die Basisdatei für einsprachige Übersetzungen herunter.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

GET /api/components/ (**string:** *project*) /
string: *component/new_template/*

Lädt die Vorlagendatei für neue Übersetzungen herunter.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

GET /api/components/ (**string:** *project*) /
string: *component/translations/*

Gibt eine Liste der Übersetzungsobjekte in der angegebenen Komponente zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **results** (*array*) – Array von Übersetzungsobjekten; siehe [GET /api/translations/ \(string:project\) / \(string:component\) / \(string:language\) /](#)

POST /api/components/ (**string:** *project*) /
string: *component/translations/*

Erstellt eine neue Übersetzung in der angegebenen Komponente.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Anfrage

- **language_code** (*string*) – Sprachcode der Übersetzung; siehe [GET /api/languages/ \(string:language\) /](#)

JSON Object-Antwort

- **result** (*object*) – Neues Übersetzungsobjekt wurde erstellt

CURL-Beispiel:

```
curl \
  -d language_code=cs \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

JSON-Anfrage-Beispiel:

```
POST /api/projects/hello/components/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/json
Authorization: Token TOKEN
Content-Length: 20
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
{"language_code": "cs"}
```

JSON-Antwort-Beispiel:

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "failing_checks": 0,
  "failing_checks_percent": 0,
  "failing_checks_words": 0,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
  "is_source": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
    "population": 1303174280,
    "name": "Czech",
    "url": "http://example.com/api/languages/cs/",
    "web_url": "http://example.com/languages/cs/"
  },
  "language_code": "cs",
  "id": 125,
  "last_author": null,
  "last_change": null,
  "share_url": "http://example.com/engage/hello/cs/",
  "total": 4,
  "total_words": 15,
  "translate_url": "http://example.com/translate/hello/weblate/cs/",
  "translated": 0,
  "translated_percent": 0.0,
  "translated_words": 0,
  "url": "http://example.com/api/translations/hello/weblate/cs/",
  "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```

GET /api/components/(string: *project*) /
string: *component*/statistics/

Gibt eine paginierte Statistik für alle Übersetzungen innerhalb der Komponente zurück.

Neu in Version 2.7.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **results** (*array*) – Array von Übersetzungsstatistikobjekten; siehe `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`

GET `/api/components/(string: project) /string: component/links/`

Gibt mit einer Komponente verknüpfte Projekte zurück.

Neu in Version 4.5.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

JSON Object-Antwort

- **projects** (*array*) – Zugeordnete Projekte; siehe `GET /api/projects/(string:project)/`

POST `/api/components/(string: project) /string: component/links/`

Ordnet Projekt einer Komponente zu.

Neu in Version 4.5.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

Formularparameter

- **string project_slug** – Projekt-Slug

DELETE `/api/components/(string: project) /string: component/links/string: project_slug/`

Aufhebung der Zuordnung eines Projekts zu einer Komponente.

Neu in Version 4.5.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **project_slug** (*string*) – Slug des zu entfernenden Projekts

1.12.9 Übersetzungen

GET `/api/translations/`

Gibt eine Liste der Übersetzungen zurück.

Siehe auch:

Die Objektattribute von Übersetzungen sind unter `GET /api/translations/(string:project)/(string:component)/(string:language)/` dokumentiert.

GET `/api/translations/(string: project) /string: component/string: language/`

Gibt Informationen über eine Übersetzung zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL

- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

JSON Object-Antwort

- **component** (*object*) – Komponentenobjekt; siehe `GET /api/components/(string:project)/(string:component)/`
- **failing_checks** (*int*) – Anzahl der Zeichenketten mit fehlgeschlagenen Qualitätsprüfungen
- **failing_checks_percent** (*float*) – Prozentsatz der Zeichenketten mit fehlgeschlagenen Qualitätsprüfungen
- **failing_checks_words** (*int*) – Anzahl der Wörter mit fehlgeschlagenen Qualitätsprüfungen
- **filename** (*string*) – Dateiname der Übersetzung
- **fuzzy** (*int*) – Anzahl der fragwürdigen (zur Bearbeitung markierten) Zeichenketten
- **fuzzy_percent** (*float*) – Prozentsatz der fragwürdigen (zur Bearbeitung markierten) Zeichenketten
- **fuzzy_words** (*int*) – Anzahl der Wörter in fragwürdigen (zur Bearbeitung markierten) Zeichenketten
- **have_comment** (*int*) – Anzahl der Zeichenketten mit Kommentar
- **have_suggestion** (*int*) – Anzahl der Zeichenketten mit Übersetzungsvorschlag
- **is_template** (*boolean*) – Ob die Übersetzung eine einsprachige Basis hat
- **language** (*object*) – Ausgangsspracheobjekt; siehe `GET /api/languages/(string:language)/`
- **language_code** (*string*) – Im Repository verwendeter Sprachcode; dieser kann sich vom Sprachcode im Sprachobjekt unterscheiden
- **last_author** (*string*) – Name des zuletzt aktiven Übersetzers
- **last_change** (*timestamp*) – Zeitstempel der letzten Änderung
- **revision** (*string*) – Revisions-Hash der Datei
- **share_url** (*string*) – URL zum Teilen, die zur Beteiligungsseite führt
- **total** (*int*) – Gesamtzahl der Zeichenketten
- **total_words** (*int*) – Gesamtzahl der Wörter
- **translate_url** (*string*) – URL der Übersetzung
- **translated** (*int*) – Anzahl übersetzter Zeichenketten
- **translated_percent** (*float*) – Prozentsatz der übersetzten Zeichenketten
- **translated_words** (*int*) – Anzahl der übersetzten Wörter
- **repository_url** (*string*) – URL zum Repository-Status; siehe `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
- **file_url** (*string*) – URL zum Dateiobjekt; siehe `GET /api/translations/(string:project)/(string:component)/(string:language)/file/`
- **changes_list_url** (*string*) – URL zur Änderungsliste; siehe `GET /api/translations/(string:project)/(string:component)/(string:language)/changes/`

- **units_list_url** (*string*) – URL zur Zeichenkettenliste; siehe `GET /api/translations/(string:project)/(string:component)/(string:language)/units/`

JSON-Daten-Beispiel:

```
{
  "component": {
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "new_base": "",
    "project": {
      "name": "Hello",
      "slug": "hello",
      "source_language": {
        "code": "en",
        "direction": "ltr",
        "population": 159034349015,
        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
      },
      "url": "http://example.com/api/projects/hello/",
      "web": "https://weblate.org/",
      "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "slug": "weblate",
    "template": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
  },
  "failing_checks": 3,
  "failing_checks_percent": 75.0,
  "failing_checks_words": 11,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
    "population": 1303174280,
    "name": "Czech",
    "url": "http://example.com/api/languages/cs/",
    "web_url": "http://example.com/languages/cs/"
  },
  "language_code": "cs",
  "last_author": "Weblate Admin",
  "last_change": "2016-03-07T10:20:05.499",
  "revision": "7ddfafe6daaf57fc8654cc852ea6be212b015792",
  "share_url": "http://example.com/engage/hello/cs/",
  "total": 4,
  "total_words": 15,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

"translate_url": "http://example.com/translate/hello/weblate/cs/",
"translated": 4,
"translated_percent": 100.0,
"translated_words": 15,
"url": "http://example.com/api/translations/hello/weblate/cs/",
"web_url": "http://example.com/projects/hello/weblate/cs/"
}

```

DELETE `/api/translations/(string: project) /`
string: `component/string: language/`

Neu in Version 3.9.

Löscht eine Übersetzung.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

GET `/api/translations/(string: project) /`
string: `component/string: language/changes/`

Gibt eine Liste der Übersetzungsänderungen zurück. Dies ist im Wesentlichen ein übersetzungsspezifisches `GET /api/changes/`, das dieselben Parameter akzeptiert.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

JSON Object-Antwort

- **results** (*array*) – Array von Komponentenobjekten; siehe `GET /api/changes/(int:id)/`

GET `/api/translations/(string: project) /`
string: `component/string: language/units/`

Gibt eine Liste der Übersetzungseinheiten zurück.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung
- **q** (*string*) – Suchanfrage-Zeichenkette *Suche* (optional)

JSON Object-Antwort

- **results** (*array*) – Array von Komponentenobjekten; siehe `GET /api/units/(int:id)/`

POST `/api/translations/(string: project) /`
string: `component/string: language/units/`

Fügt eine neue Einheit hinzu.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL

- **language** (*string*) – Sprachcode der Übersetzung

JSON Object-Anfrage

- **key** (*string*) – Name der Übersetzungseinheit (verwendet als Schlüssel oder Kontext)
- **value** (*array*) – Ausgangszeichenketten (einzelne Zeichenkette verwenden, wenn kein Plural erstellt wird)
- **state** (*int*) – Zeichenkettenstatus; siehe [GET /api/units/\(int:id\)/](#)

JSON Object-Antwort

- **unit** (*object*) – Neu erstellte Einheit; siehe [GET /api/units/\(int:id\)/](#)

Siehe auch:

[Zeichenketten verwalten](#), [adding-new-strings](#)

POST [/api/translations/\(string: project\) /](#)
string: component/string: language/autotranslate/

Löst die automatische Übersetzung aus.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

JSON Object-Anfrage

- **mode** (*string*) – Automatischer Übersetzungsmodus
- **filter_type** (*string*) – Filtertyp der automatischen Übersetzung
- **auto_source** (*string*) – Automatische Übersetzung – mt oder others
- **component** (*string*) – Aktivieren Sie den Beitrag zum gemeinsamen Übersetzungsspeicher für das Projekt, um Zugriff auf zusätzliche Komponenten zu erhalten.
- **engines** (*array*) – Vorschläge aus automatischer Übersetzung
- **threshold** (*string*) – Scoreschwelle

GET [/api/translations/\(string: project\) /](#)
string: component/string: language/file/

Lädt die aktuelle Übersetzungsdatei, wie sie im VCS gespeichert ist (ohne den Parameter `format`) oder in ein anderes Format konvertiert wurde (siehe [Übersetzungen herunterladen](#)) herunter.

Bemerkung: Dieser API-Endpunkt verwendet eine andere Logik für die Ausgabe als der Rest der API, da er mit der gesamten Datei und nicht mit den Daten arbeitet. Die Menge der akzeptierten `format`-Parameter unterscheidet sich und ohne einen solchen Parameter erhalten Sie die Übersetzungsdatei, wie sie im VCS gespeichert ist.

Abfrageparameter

- **format** – Zu verwendendes Dateiformat; wenn nicht angegeben, erfolgt keine Formatkonvertierung; unterstützte Dateiformate: po, mo, xiff, xiff11, tbx, tmx, csv, xlsx, json, aresource, strings
- **q** (*string*) – Filter für heruntergeladene Zeichenketten, siehe [search](#), nur anwendbar, wenn eine Konvertierung stattfindet (`format` ist angegeben).

Parameter

- **project** (*string*) – Kurzer Projekt-URL

- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

POST /api/translations/(**string**: *project*) /
string: *component*/**string**: *language*/**file**/

Lädt eine neue Datei mit Übersetzungen hoch.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

Formularparameter

- **string conflicts** – Wie mit Konflikten umgegangen wird (ignore, replace-translated oder replace-approved)
- **file file** – Hochgeladene Datei
- **string email** – Autor-E-Mail
- **string author** – Autor-Name
- **string method** – Upload-Methode (translate, approve, suggest, fuzzy, replace, source, add), siehe [Importmethoden](#)
- **string fuzzy** – Verarbeitung von fragwürdigen (zur Bearbeitung markierte) Zeichenketten (*leer*, process, approve)

CURL-Beispiel:

```
curl -X POST \
  -F file=@strings.xml \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/translations/hello/android/cs/file/
```

GET /api/translations/(**string**: *project*) /
string: *component*/**string**: *language*/**repository**/

Gibt Informationen über den Status des VCS-Repositorys zurück.

Die Antwort ist dieselbe wie bei `GET /api/components/(string:project)/(string:component)/repository/`.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

POST /api/translations/(**string**: *project*) /
string: *component*/**string**: *language*/**repository**/

Führt eine bestimmte Operation im VCS-Repository durch.

Siehe `POST /api/projects/(string:project)/repository/` für die Dokumentation.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

JSON Object-Anfrage

- **operation** (*string*) – Auszuführende Operation: eine von push, pull, commit, reset, cleanup

JSON Object-Antwort

- **result** (*boolean*) – Ergebnis der Operation

GET /api/translations/ (**string**: *project*) /
string: *component*/**string**: *language/statistics/*

Gibt detaillierte Übersetzungsstatistiken zurück.

Neu in Version 2.7.

Parameter

- **project** (*string*) – Kurzer Projekt-URL
- **component** (*string*) – Kurzer Komponenten-URL
- **language** (*string*) – Sprachcode der Übersetzung

JSON Object-Antwort

- **code** (*string*) – Sprachcode
- **failing** (*int*) – Anzahl der fehlgeschlagenen Qualitätsprüfungen
- **failing_percent** (*float*) – Prozentsatz der fehlgeschlagenen Qualitätsprüfungen
- **fuzzy** (*int*) – Anzahl der fragwürdigen (zur Bearbeitung markierten) Zeichenketten
- **fuzzy_percent** (*float*) – Prozentsatz der fragwürdigen (zur Bearbeitung markierten) Zeichenketten
- **total_words** (*int*) – Gesamtzahl der Wörter
- **translated_words** (*int*) – Anzahl der übersetzten Wörter
- **last_author** (*string*) – Name des zuletzt aktiven Übersetzers
- **last_change** (*timestamp*) – Datum der letzten Änderung
- **name** (*string*) – Sprachename
- **total** (*int*) – Gesamtzahl der Zeichenketten
- **translated** (*int*) – Anzahl übersetzter Zeichenketten
- **translated_percent** (*float*) – Prozentsatz der übersetzten Zeichenketten
- **url** (*string*) – URL für den Zugriff auf die Übersetzung (Beteiligungs-URL)
- **url_translate** (*string*) – URL für den Zugriff auf die Übersetzung (echte Übersetzungs-URL)

1.12.10 Speicher

Neu in Version 4.14.

GET /api/memory/

Gibt eine Liste der Speicherergebnisse zurück.

DELETE /api/memory/ (**int**: *memory_object_id*) /

Löscht ein Speicherobjekt

Parameter

- **memory_object_id** – Speicherobjekt-ID

1.12.11 Einheiten

Eine *Einheit* ist ein einzelner Teil einer Übersetzung, der eine Ausgangszeichenkette mit einer entsprechenden übersetzten Zeichenkette verbindet und auch einige zugehörige Metadaten enthält. Der Begriff stammt aus dem „Translate-Toolkit“ <<http://docs.translatehouse.org/projects/translate-toolkit/en/latest/api/storage.html#translate.storage.base.TranslationUnit>>`_ und XLIFF.

Neu in Version 2.10.

GET `/api/units/`

Gibt eine Liste der Übersetzungseinheiten zurück.

Parameter

- **q** (*string*) – Suchanfrage-Zeichenkette *Suche* (optional)

Siehe auch:

Die Objektattribute von Einheiten sind unter `GET /api/units/(int:id)/` dokumentiert.

GET `/api/units/(int: id) /`

Geändert in Version 4.3: `target` und `source` sind jetzt Arrays, um Zeichenketten mit Pluralformen richtig zu behandeln.

Gibt Informationen über die Übersetzungseinheit zurück.

Parameter

- **id** (*int*) – Einheitenkennung

JSON Object-Antwort

- **translation** (*string*) – URL eines zugehörigen Übersetzungsobjekts
- **source** (*array*) – Ausgangszeichenkette
- **previous_source** (*string*) – Vorherige Ausgangszeichenkette, die für Fuzzy-Matching verwendet wurde
- **target** (*array*) – Zielzeichenkette
- **id_hash** (*string*) – Eindeutige Kennung der Einheit
- **content_hash** (*string*) – Eindeutige Kennung der Ausgangszeichenkette
- **location** (*string*) – Position der Einheit im Quellcode
- **context** (*string*) – Kontext der Übersetzungseinheit
- **note** (*string*) – Anmerkung zur Übersetzungseinheit
- **flags** (*string*) – Markierung der Übersetzungseinheit
- **labels** (*array*) – Labels der Übersetzungseinheit, verfügbar in der Ausgangseinheit
- **state** (*int*) – Status der Einheit, 0 – nicht übersetzt, 10 – Bearbeitung erforderlich, 20 – übersetzt, 30 – genehmigt, 100 – schreibgeschützt
- **fuzzy** (*boolean*) – Ob die Einheit fragwürdig oder zur Überprüfung markiert ist
- **translated** (*boolean*) – Ob die Einheit übersetzt wird
- **approved** (*boolean*) – Ob die Übersetzung genehmigt wird
- **position** (*int*) – Position der Einheit in der Übersetzungsdatei
- **has_suggestion** (*boolean*) – Ob die Einheit Vorschläge hat
- **has_comment** (*boolean*) – Ob die Einheit Kommentare hat
- **has_failing_check** (*boolean*) – Ob die Einheit fehlerhafte Prüfungen aufweist
- **num_words** (*int*) – Anzahl der Ausgangswörter

- **priority** (*int*) – Übersetzungspriorität; 100 ist Standard
- **id** (*int*) – Einheitenkennung
- **explanation** (*string*) – Erläuterung der Zeichenkette, verfügbar in den Ausgangseinheiten, siehe [Zusätzliche Informationen über Ausgangszeichenkette](#)
- **extra_flags** (*string*) – Zusätzliche Markierungen der Zeichenkette, verfügbar in den Ausgangseinheiten, siehe [Anpassen des Verhaltens mit Markierungen](#)
- **web_url** (*string*) – URL, unter der die Einheit bearbeitet werden kann
- **source_unit** (*string*) – Link zur Ausgangseinheit; siehe `GET /api/units/(int:id)/`
- **pending** (*boolean*) – Ob die Einheit zum Schreiben aussteht
- **timestamp** (*timestamp*) – Alter der Zeichenkette

PATCH /api/units/(int: id) /

Neu in Version 4.3.

Führt eine teilweise Aktualisierung der Übersetzungseinheit durch.

Parameter

- **id** (*int*) – Einheitenkennung

JSON Object-Anfrage

- **state** (*int*) – Status der Einheit, 0 – nicht übersetzt, 10 – Bearbeitung erforderlich, 20 – übersetzt, 30 – genehmigt (Überprüfungsablauf muss aktiviert sein, siehe [Zugehörige Prüfer](#))
- **target** (*array*) – Zielzeichenkette
- **explanation** (*string*) – Erläuterung der Zeichenkette, verfügbar in den Ausgangseinheiten, siehe [Zusätzliche Informationen über Ausgangszeichenkette](#)
- **extra_flags** (*string*) – Zusätzliche Markierungen der Zeichenkette, verfügbar in den Ausgangseinheiten, siehe [Anpassen des Verhaltens mit Markierungen](#)

JSON Object-Antwort

- **labels** (*array*) – Labels, verfügbar in den Ausgangseinheiten

PUT /api/units/(int: id) /

Neu in Version 4.3.

Führt ein vollständiges Update der Übersetzungseinheit durch.

Parameter

- **id** (*int*) – Einheitenkennung

JSON Object-Anfrage

- **state** (*int*) – Status der Einheit, 0 – nicht übersetzt, 10 – Bearbeitung erforderlich, 20 – übersetzt, 30 – genehmigt (Überprüfungsablauf muss aktiviert sein, siehe [Zugehörige Prüfer](#))
- **target** (*array*) – Zielzeichenkette
- **explanation** (*string*) – Erläuterung der Zeichenkette, verfügbar in den Ausgangseinheiten, siehe [Zusätzliche Informationen über Ausgangszeichenkette](#)
- **extra_flags** (*string*) – Zusätzliche Markierungen der Zeichenkette, verfügbar in den Ausgangseinheiten, siehe [Anpassen des Verhaltens mit Markierungen](#)

JSON Object-Antwort

- **labels** (*array*) – Labels, verfügbar in den Ausgangseinheiten

DELETE `/api/units/(int: id) /`

Neu in Version 4.3.

Löscht eine Übersetzungseinheit.

Parameter

- **id** (*int*) – Einheitenkennung

1.12.12 Änderungen

Neu in Version 2.10.

GET `/api/changes/`

Geändert in Version 4.1: Die Filterung von Änderungen wurde mit der Version 4.1 eingeführt.

Gibt eine Liste der Übersetzungsänderungen zurück.

Siehe auch:

Die Objektattribute von Änderungen sind unter `GET /api/changes/(int:id) /` dokumentiert.

Abfrageparameter

- **user** (*string*) – Benutzername des zu filternden Benutzers
- **action** (*int*) – Aktion zum Filtern, kann mehrfach verwendet werden
- **timestamp_after** (*timestamp*) – ISO 8601-formatierter Zeitstempel zur Auflistung von Änderungen nach
- **timestamp_before** (*timestamp*) – ISO 8601-formatierter Zeitstempel zur Auflistung von Änderungen vor

GET `/api/changes/(int: id) /`

Gibt Informationen über Übersetzungsänderungen zurück.

Parameter

- **id** (*int*) – Änderungs-ID

JSON Object-Antwort

- **unit** (*string*) – URL eines zugehörigen Einheitenobjekts
- **translation** (*string*) – URL eines zugehörigen Übersetzungsobjekts
- **component** (*string*) – URL eines zugehörigen Komponentenobjekts
- **user** (*string*) – URL eines zugehörigen Benutzerobjekts
- **author** (*string*) – URL eines zugehörigen Autorenobjekts
- **timestamp** (*timestamp*) – Zeitstempel des Ereignisses
- **action** (*int*) – Numerische Kennzeichnung der Aktion
- **action_name** (*string*) – Textbeschreibung der Aktion
- **target** (*string*) – Änderungstext oder -detail des Ereignisses
- **id** (*int*) – Änderungskennung

1.12.13 Bildschirmfotos

Neu in Version 2.14.

GET /api/screenshots/

Gibt eine Liste der Zeichenketteninformationen-Bildschirmfotos zurück.

Siehe auch:

Die Objektattribute von Bildschirmfotos sind unter `GET /api/screenshots/(int:id)/` dokumentiert.

GET /api/screenshots/(int: id) /

Gibt Informationen über Bildschirmfotos zurück.

Parameter

- **id** (*int*) – Bildschirmfoto-ID

JSON Object-Antwort

- **name** (*string*) – Name des Bildschirmfotos
- **component** (*string*) – URL eines zugehörigen Komponentenobjekts
- **file_url** (*string*) – URL zum Herunterladen einer Datei; siehe `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – Link zur zugehörigen Information der Ausgangszeichenkette; siehe `GET /api/units/(int:id)/`

GET /api/screenshots/(int: id) /file/

Lädt das Bildschirmfoto herunter.

Parameter

- **id** (*int*) – Bildschirmfoto-ID

POST /api/screenshots/(int: id) /file/

Ersetzt das Bildschirmfoto.

Parameter

- **id** (*int*) – Bildschirmfoto-ID

Formularparameter

- **file image** – Hochgeladene Datei

CURL-Beispiel:

```
curl -X POST \
  -F image=@image.png \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/screenshots/1/file/
```

POST /api/screenshots/(int: id) /units/

Ordnet Ausgangszeichenkette einem Bildschirmfoto zu.

Parameter

- **id** (*int*) – Bildschirmfoto-ID

Formularparameter

- **string unit_id** – Einheitenkennung

JSON Object-Antwort

- **name** (*string*) – Name des Bildschirmfotos

- **translation** (*string*) – URL eines zugehörigen Übersetzungsobjekts
- **file_url** (*string*) – URL zum Herunterladen einer Datei; siehe `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – Link zur zugehörigen Information der Ausgangszeichenkette; siehe `GET /api/units/(int:id)/`

DELETE `/api/screenshots/(int: id)/units/`

int: *unit_id*

Entfernt die Verknüpfung der Ausgangszeichenkette mit dem Bildschirmfoto.

Parameter

- **id** (*int*) – Bildschirmfoto-ID
- **unit_id** – Ausgangszeichenkette-Einheitenkennung

POST `/api/screenshots/`

Erstellt ein neues Bildschirmfoto.

Formularparameter

- **file image** – Hochgeladene Datei
- **string name** – Name des Bildschirmfotos
- **string project_slug** – Projekt-Slug
- **string component_slug** – Komponenten-Slug
- **string language_code** – Sprachcode

JSON Object-Antwort

- **name** (*string*) – Name des Bildschirmfotos
- **component** (*string*) – URL eines zugehörigen Komponentenobjekts
- **file_url** (*string*) – URL zum Herunterladen einer Datei; siehe `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – Link zur zugehörigen Information der Ausgangszeichenkette; siehe `GET /api/units/(int:id)/`

PATCH `/api/screenshots/(int: id) /`

Bearbeitet Teilinformationen zum Bildschirmfoto.

Parameter

- **id** (*int*) – Bildschirmfoto-ID

JSON Object-Antwort

- **name** (*string*) – Name des Bildschirmfotos
- **component** (*string*) – URL eines zugehörigen Komponentenobjekts
- **file_url** (*string*) – URL zum Herunterladen einer Datei; siehe `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – Link zur zugehörigen Information der Ausgangszeichenkette; siehe `GET /api/units/(int:id)/`

PUT `/api/screenshots/(int: id) /`

Bearbeitet alle Informationen zum Bildschirmfoto.

Parameter

- **id** (*int*) – Bildschirmfoto-ID

JSON Object-Antwort

- **name** (*string*) – Name des Bildschirmfotos
- **component** (*string*) – URL eines zugehörigen Komponentenobjekts
- **file_url** (*string*) – URL zum Herunterladen einer Datei; siehe `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – Link zur zugehörigen Information der Ausgangszeichenkette; siehe `GET /api/units/(int:id)/`

DELETE `/api/screenshots/(int: id) /`

Löscht das Bildschirmfoto.

Parameter

- **id** (*int*) – Bildschirmfoto-ID

1.12.14 Erweiterungen

Neu in Version 4.4.1.

GET `/api/addons/`

Gibt eine Liste der Erweiterungen zurück.

Siehe auch:

Die Objektattribute von Erweiterungen sind unter `GET /api/addons/(int:id)/` dokumentiert.

GET `/api/addons/(int: id) /`

Gibt Informationen über Erweiterungen zurück.

Parameter

- **id** (*int*) – Erweiterungs-ID

JSON Object-Antwort

- **name** (*string*) – Name der Erweiterung
- **component** (*string*) – URL eines zugehörigen Komponentenobjekts
- **configuration** (*object*) – Optionale Konfiguration der Erweiterung

Siehe auch:

[Erweiterungen](#)

POST `/api/components/(string: project) /`
`string: component/addons/`

Erstellt eine neue Erweiterung.

Parameter

- **project_slug** (*string*) – Projekt-Slug
- **component_slug** (*string*) – Komponenten-Slug

JSON Object-Anfrage

- **name** (*string*) – Name der Erweiterung
- **configuration** (*object*) – Optionale Konfiguration der Erweiterung

PATCH `/api/addons/(int: id) /`

Bearbeitet Teilinformationen zur Erweiterung.

Parameter

- **id** (*int*) – Erweiterungs-ID

JSON Object-Antwort

- **configuration** (*object*) – Optionale Konfiguration der Erweiterung

PUT /api/addons/ (int: id) /

Bearbeitet alle Informationen zur Erweiterung.

Parameter

- **id** (*int*) – Erweiterungs-ID

JSON Object-Antwort

- **configuration** (*object*) – Optionale Konfiguration der Erweiterung

DELETE /api/addons/ (int: id) /

Löscht die Erweiterung.

Parameter

- **id** (*int*) – Erweiterungs-ID

1.12.15 Komponentenlisten

Neu in Version 4.0.

GET /api/component-lists/

Gibt eine Liste der Komponentenlisten zurück.

Siehe auch:

Die Objektattribute von Komponentenlisten sind unter [GET /api/component-lists/\(str:slug\)/](#) dokumentiert.

GET /api/component-lists/ (str: slug) /

Gibt Informationen über die Komponentenliste zurück.

Parameter

- **slug** (*string*) – Komponentenliste-Slug

JSON Object-Antwort

- **name** (*string*) – Name einer Komponentenliste
- **slug** (*string*) – Slug einer Komponentenliste
- **show_dashboard** (*boolean*) – Ob es auf einer Übersichtsseite angezeigt werden soll
- **components** (*array*) – Link zu den zugeordneten Komponenten; siehe [GET /api/components/\(string:project\)/\(string:component\)/](#)
- **auto_assign** (*array*) – Automatische Zuweisungsregeln

PUT /api/component-lists/ (str: slug) /

Ändert die Parameter der Komponentenliste.

Parameter

- **slug** (*string*) – Komponentenliste-Slug

JSON Object-Anfrage

- **name** (*string*) – Name einer Komponentenliste
- **slug** (*string*) – Slug einer Komponentenliste
- **show_dashboard** (*boolean*) – Ob es auf einer Übersichtsseite angezeigt werden soll

PATCH /api/component-lists/ (str: slug) /

Ändert die Parameter der Komponentenliste.

Parameter

- **slug** (string) – Komponentenliste-Slug

JSON Object-Anfrage

- **name** (string) – Name einer Komponentenliste
- **slug** (string) – Slug einer Komponentenliste
- **show_dashboard** (boolean) – Ob es auf einer Übersichtsseite angezeigt werden soll

DELETE /api/component-lists/ (str: slug) /

Löscht die Komponentenliste.

Parameter

- **slug** (string) – Komponentenliste-Slug

POST /api/component-lists/ (str: slug) /components/

Ordnet Komponente einer Komponentenliste zu.

Parameter

- **slug** (string) – Komponentenliste-Slug

Formularparameter

- **string component_id** – Komponenten-ID

DELETE /api/component-lists/ (str: slug) /components/
str: component_slug

Hebt die Zuordnung einer Komponente zur Komponentenliste auf.

Parameter

- **slug** (string) – Komponentenliste-Slug
- **component_slug** (string) – Komponenten-Slug

1.12.16 Glossar

Geändert in Version 4.5: Glossare werden nun als reguläre Komponenten, Übersetzungen und Zeichenketten gespeichert, bitte verwenden Sie stattdessen die entsprechende API.

1.12.17 Aufgaben

Neu in Version 4.4.

GET /api/tasks/

Eine Auflistung der Aufgaben ist derzeit nicht verfügbar.

GET /api/tasks/ (str: uuid) /

Gibt Informationen über eine Aufgabe zurück.

Parameter

- **uuid** (string) – Aufgaben-UUID

JSON Object-Antwort

- **completed** (boolean) – Ob die Aufgabe abgeschlossen wurde
- **progress** (int) – Aufgabenfortschritt in Prozent

- **result** (*object*) – Details zum Aufgabenergebnis oder -fortschritt
- **log** (*string*) – Aufgabenprotokoll

1.12.18 Metriken

GET `/api/metrics/`

Gibt Server-Metriken zurück.

JSON Object-Antwort

- **units** (*int*) – Anzahl der Einheiten
- **units_translated** (*int*) – Anzahl der übersetzten Einheiten
- **users** (*int*) – Anzahl der Benutzer
- **changes** (*int*) – Anzahl der Änderungen
- **projects** (*int*) – Anzahl der Projekte
- **components** (*int*) – Anzahl der Komponenten
- **translations** (*int*) – Anzahl der Übersetzungen
- **languages** (*int*) – Anzahl der verwendeten Sprachen
- **checks** (*int*) – Anzahl der ausgelösten Qualitätsprüfungen
- **configuration_errors** (*int*) – Anzahl der Konfigurationsfehler
- **suggestions** (*int*) – Anzahl der ausstehenden Übersetzungsvorschläge
- **celery_queues** (*object*) – Längen der Celery-Warteschlangen, siehe [Hintergrund-aufgaben mit Celery](#)
- **name** (*string*) – Konfigurierter Servername

1.12.19 Benachrichtigungs-Hooks

Benachrichtigungs-Hooks ermöglichen es externen Anwendungen, Weblate zu benachrichtigen, dass das VCS-Repository aktualisiert wurde.

Sie können Repository-Endpunkte für Projekte, Komponenten und Übersetzungen verwenden, um einzelne Repositories zu aktualisieren; siehe `POST /api/projects/(string:project)/repository/` für die Dokumentation.

GET `/hooks/update/(string: project) /`
string: *component/*

Veraltet ab Version 2.6: Bitte verwenden Sie stattdessen `POST /api/components/(string:project)/(string:component)/repository/`, das mit der Authentifizierung für ACL-begrenzte Projekte korrekt funktioniert.

Löst die Aktualisierung einer Komponente aus (Abruf aus dem VCS und Scannen nach Übersetzungsänderungen).

GET `/hooks/update/(string: project) /`

Veraltet ab Version 2.6: Bitte verwenden Sie stattdessen `POST /api/projects/(string:project)/repository/`, das mit der Authentifizierung für ACL-begrenzte Projekte korrekt funktioniert.

Löst die Aktualisierung aller Komponenten in einem Projekt aus (Abruf aus dem VCS und Scannen nach Übersetzungsänderungen).

POST /hooks/github/

Spezieller Hook für die Behandlung von GitHub-Benachrichtigungen und die automatische Aktualisierung passender Komponenten.

Bemerkung: GitHub bietet direkte Unterstützung für die Benachrichtigung von Weblate: Aktivieren Sie den Weblate-Service-Hook in den Repository-Einstellungen und setzen Sie die URL auf die URL Ihrer Weblate-Installation.

Siehe auch:

Automatisches Empfangen von Änderungen von GitHub

Für Anweisungen zur Einrichtung der GitHub-Integration

<https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks>

Allgemeine Informationen zu GitHub-Webhooks

ENABLE_HOOKS

Zum Aktivieren von Hooks für das gesamte Weblate

POST /hooks/gitlab/

Spezieller Hook für die Behandlung von GitLab-Benachrichtigungen und die automatische Aktualisierung passender Komponenten.

Siehe auch:

Automatischer Empfang von Änderungen von GitLab

Für Anweisungen zur Einrichtung der GitLab-Integration

<https://docs.gitlab.com/ee/user/project/integrations/webhooks.html>

Allgemeine Informationen zu GitLab-Webhooks

ENABLE_HOOKS

Zum Aktivieren von Hooks für das gesamte Weblate

POST /hooks/bitbucket/

Spezieller Hook für die Behandlung von Bitbucket-Benachrichtigungen und die automatische Aktualisierung passender Komponenten.

Siehe auch:

Automatischer Empfang von Änderungen von Bitbucket

Für Anweisungen zur Einrichtung der Bitbucket-Integration

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/>

Allgemeine Informationen zu Bitbucket-Webhooks

ENABLE_HOOKS

Zum Aktivieren von Hooks für das gesamte Weblate

POST /hooks/pagure/

Neu in Version 3.3.

Spezieller Hook für die Behandlung von Pagure-Benachrichtigungen und die automatische Aktualisierung passender Komponenten.

Siehe auch:

Automatischer Empfang von Änderungen von Pagure

Für Anweisungen zur Einrichtung der Pagure-Integration

https://docs.pagure.org/pagure/usage/using_webhooks.html

Allgemeine Informationen zu Pagure-Webhooks

ENABLE_HOOKS

Zum Aktivieren von Hooks für das gesamte Weblate

POST /hooks/azure/

Neu in Version 3.8.

Spezieller Hook für die Behandlung von Azure-DevOps-Benachrichtigungen und die automatische Aktualisierung passender Komponenten.

Bemerkung: Bitte stellen Sie sicher, dass *Resource details to send* auf *Alle* gesetzt ist, da Weblate sonst nicht in der Lage ist, Ihr Azure-Repository zu finden.

Siehe auch:

Automatischer Empfang von Änderungen von Azure Repos

Für Anweisungen zur Einrichtung der Azure-Integration

https:

[//learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops](https://learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops)

Allgemeine Informationen zu Azure-DevOps-Webhooks

ENABLE_HOOKS

Zum Aktivieren von Hooks für das gesamte Weblate

POST /hooks/gitea/

Neu in Version 3.9.

Spezieller Hook für die Behandlung von Gitea-Webhook-Benachrichtigungen und die automatische Aktualisierung passender Komponenten.

Siehe auch:

Automatischer Empfang von Änderungen von Gitea Repos

Für Anweisungen zur Einrichtung der Gitea-Integration

https://docs.gitea.io/en-us/webhooks/

Allgemeine Informationen zu Gitea-Webhooks

ENABLE_HOOKS

Zum Aktivieren von Hooks für das gesamte Weblate

POST /hooks/gitee/

Neu in Version 3.9.

Spezieller Hook für die Behandlung von Gitee-Webhook-Benachrichtigungen und die automatische Aktualisierung passender Komponenten.

Siehe auch:

Automatischer Empfang von Änderungen von Gitee Repos

Für Anweisungen zur Einrichtung der Gitee-Integration

https://gitee.com/help/categories/40

Allgemeine Informationen zu Gitee-Webhooks

ENABLE_HOOKS

Zum Aktivieren von Hooks für das gesamte Weblate

1.12.20 Exporte

Weblate bietet verschiedene Exportmöglichkeiten, mit denen Sie die Daten weiterverarbeiten können.

GET `/exports/stats/(string: project) /`
`string: component/`

Abfrageparameter

- **format** (*string*) – Ausgabeformat: entweder json oder csv

Veraltet ab Version 2.6: Bitte verwenden Sie stattdessen `GET /api/components/(string:project)/(string:component)/statistics/` und `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`; dies ermöglicht auch den Zugriff auf ACL-gesteuerte Projekte.

Ruft Statistiken für die angegebene Komponente im angegebenen Format ab.

Beispielanfrage:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Beispielantwort:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

[
  {
    "code": "cs",
    "failing": 0,
    "failing_percent": 0.0,
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "last_author": "Michal Čihař",
    "last_change": "2012-03-28T15:07:38+00:00",
    "name": "Czech",
    "total": 436,
    "total_words": 15271,
    "translated": 436,
    "translated_percent": 100.0,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/cs/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
  },
  {
    "code": "nl",
    "failing": 21,
    "failing_percent": 4.8,
    "fuzzy": 11,
    "fuzzy_percent": 2.5,
    "last_author": null,
    "last_change": null,
    "name": "Dutch",
    "total": 436,
    "total_words": 15271,
    "translated": 319,
    "translated_percent": 73.2,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/nl/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/nl/"
  }
]
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

},
{
  "code": "el",
  "failing": 11,
  "failing_percent": 2.5,
  "fuzzy": 21,
  "fuzzy_percent": 4.8,
  "last_author": null,
  "last_change": null,
  "name": "Greek",
  "total": 436,
  "total_words": 15271,
  "translated": 312,
  "translated_percent": 71.6,
  "translated_words": 3201,
  "url": "http://hosted.weblate.org/engage/weblate/el/",
  "url_translate": "http://hosted.weblate.org/projects/weblate/main/el/"
}
]

```

1.12.21 RSS-Feeds

Änderungen an Übersetzungen werden in RSS-Feeds exportiert.

GET `/exports/rss/(string: project) /`
string: `component/string: language/`

Ruft den RSS-Feed mit den letzten Änderungen für eine Übersetzung ab.

GET `/exports/rss/(string: project) /`
string: `component/`

Ruft den RSS-Feed mit den letzten Änderungen für eine Komponente ab.

GET `/exports/rss/(string: project) /`

Ruft den RSS-Feed mit den letzten Änderungen für ein Projekt ab.

GET `/exports/rss/language/(string: language) /`

Ruft den RSS-Feed mit den letzten Änderungen für eine Sprache ab.

GET `/exports/rss/`

Ruft den RSS-Feed mit den letzten Änderungen für die Weblate-Instanz ab.

Siehe auch:

[RSS auf Wikipedia](#)

1.13 Weblate-Client

Neu in Version 2.7: There has been full wlc utility support ever since Weblate 2.7. If you are using an older version some incompatibilities with the API might occur.

1.13.1 Installation

The Weblate Client is shipped separately and includes the Python module. To use the commands below, you need to install `wlc`:

```
pip install wlc
```

1.13.2 Verwendung von Docker

The Weblate Client is also available as a Docker image.

The image is published on Docker Hub: <https://hub.docker.com/r/weblate/wlc>

Installieren:

```
docker pull weblate/wlc
```

Der Docker-Container verwendet die Standardeinstellungen von Weblate und verbindet sich mit der auf localhost bereitgestellten API. Die API-URL und der API_KEY können über die von Weblate akzeptierten Argumente konfiguriert werden.

The command to launch the container uses the following syntax:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Beispiel:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

You might want to pass your *Konfigurationsdateien* to the Docker container, the easiest approach is to add your current directory as `/home/weblate` volume:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Erste Schritte

The `wlc` configuration is stored in `~/.config/weblate` (see *Konfigurationsdateien* for other locations), please create it to match your environment:

```
[weblate]
url = https://hosted.weblate.org/api/

[keys]
https://hosted.weblate.org/api/ = APIKEY
```

You can then invoke commands on the default server:

```
wlc ls
wlc commit sandbox/hello-world
```

Siehe auch:

Konfigurationsdateien

1.13.4 Kurzfassung

```
wlc [arguments] <command> [options]
```

Commands actually indicate which operation should be performed.

1.13.5 Beschreibung

Weblate Client is a Python library and command-line utility to manage Weblate remotely using *Weblates REST API*. The command-line utility can be invoked as **wlc** and is built-in on *wlc*.

Argumente

Das Programm akzeptiert die folgenden Argumente, die das Ausgabeformat oder die zu verwendende Weblate-Instanz definieren. Diese müssen vor jedem Befehl eingegeben werden.

--format {csv,json,text,html}

Specify the output format.

--url URL

Specify the API URL. Overrides any value found in the configuration file, see *Konfigurationsdateien*. The URL should end with `/api/`, for example `https://hosted.weblate.org/api/`.

--key KEY

Specify the API user key to use. Overrides any value found in the configuration file, see *Konfigurationsdateien*. You can find your key in your profile on Weblate.

--config PATH

Overrides the configuration file path, see *Konfigurationsdateien*.

--config-section SECTION

Overrides configuration file section in use, see *Konfigurationsdateien*.

Befehle

The following commands are available:

version

Prints the current version.

list-languages

Lists used languages in Weblate.

list-projects

Lists projects in Weblate.

list-components

Lists components in Weblate.

list-translations

Lists translations in Weblate.

show

Shows Weblate object (translation, component or project).

ls

Lists Weblate object (translation, component or project).

commit

Commits changes made in a Weblate object (translation, component or project).

pull

Pulls remote repository changes into Weblate object (translation, component or project).

push

Pushes Weblate object changes into remote repository (translation, component or project).

reset

Neu in Version 0.7: Unterstützt seit wlc 0.7.

Resets changes in Weblate object to match remote repository (translation, component or project).

cleanup

Neu in Version 0.9: Unterstützt seit wlc 0.9.

Removes any untracked changes in a Weblate object to match the remote repository (translation, component or project).

repo

Displays repository status for a given Weblate object (translation, component or project).

stats

Displays detailed statistics for a given Weblate object (translation, component or project).

lock-status

Neu in Version 0.5: Unterstützt seit wlc 0.5.

Displays lock status.

lock

Neu in Version 0.5: Unterstützt seit wlc 0.5.

Locks component from further translation in Weblate.

unlock

Neu in Version 0.5: Unterstützt seit wlc 0.5.

Unlocks translation of Weblate component.

changes

Neu in Version 0.7: Unterstützt seit wlc 0.7 und Weblate 2.10.

Displays changes for a given object.

download

Neu in Version 0.7: Unterstützt seit wlc 0.7.

Downloads a translation file.

--convert

Converts file format, if unspecified no conversion happens on the server and the file is downloaded as is to the repository.

--output

Specifies file to save output in, if left unspecified it is printed to stdout.

upload

Neu in Version 0.9: Unterstützt seit wlc 0.9.

Uploads a translation file.

--overwrite

Beim Hochladen vorhandene Übersetzungen überschreiben.

--input

File from which content is read, if left unspecified it is read from stdin.

--method

Upload method to use, see *Importmethoden*.

--fuzzy

Verarbeitung von fragwürdigen (zur Bearbeitung markierte) Zeichenketten (*leer*, *process*, *approve*)

--author-name

Author name, to override currently authenticated user

--author-email

Author e-mail, to override currently authenticated user

Hinweis: You can get more detailed information on invoking individual commands by passing `--help`, for example:
`wlc ls --help`.

1.13.6 Konfigurationsdateien

.weblate, .weblate.ini, weblate.ini

Geändert in Version 1.6: Die Dateien mit der Erweiterung *.ini* werden ebenfalls akzeptiert.

Pro Projekt-Konfigurationsdatei

C:\Users\NAME\AppData\weblate.ini

Neu in Version 1.6.

Benutzer-Konfigurationsdatei unter Windows.

~/.config/weblate

Benutzer-Konfigurationsdatei

/etc/xdg/weblate

Systemweite Konfigurationsdatei

The program follows the XDG specification, so you can adjust placement of config files by environment variables `XDG_CONFIG_HOME` or `XDG_CONFIG_DIRS`. On Windows APPDATA directory is preferred location for the configuration file.

Following settings can be configured in the `[weblate]` section (you can customize this by *--config-section*):

key

API-Schlüssel für den Zugang zu Weblate.

url

API server URL, defaults to `http://127.0.0.1:8000/api/`.

translation

Pfad zur Standardübersetzung – Komponente oder Projekt.

Die Konfigurationsdatei ist eine INI-Datei, zum Beispiel:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Zusätzlich können API-Schlüssel im Abschnitt `[keys]` gespeichert werden:

[keys]

```
https://hosted.weblate.org/api/ = APIKEY
```

This allows you to store keys in your personal settings, while using the `.weblate` configuration in the VCS repository so that `wlc` knows which server it should talk to.

1.13.7 Beispiele

Aktuelle Programmversion drucken:

```
$ wlc version
version: 0.1
```

Alle Projekte auflisten:

```
$ wlc list-projects
name: Hello
slug: hello
url: http://example.com/api/projects/hello/
web: https://weblate.org/
web_url: http://example.com/projects/hello/
```

Übersetzungsdatei hochladen:

```
$ wlc upload project/component/language --input /tmp/hello.po
```

Sie können auch angeben, an welchem Projekt `wlc` arbeiten soll:

```
$ cat .weblate
[weblate]
url = https://hosted.weblate.org/api/
translation = weblate/application

$ wlc show
branch: main
file_format: po
source_language: en
filemask: weblate/locale/*/LC_MESSAGES/django.po
git_export: https://hosted.weblate.org/git/weblate/application/
license: GPL-3.0+
license_url: https://spdx.org/licenses/GPL-3.0+
name: Application
new_base: weblate/locale/django.pot
project: weblate
repo: git://github.com/WeblateOrg/weblate.git
slug: application
template:
url: https://hosted.weblate.org/api/components/weblate/application/
vcs: git
web_url: https://hosted.weblate.org/projects/weblate/application/
```

Mit diesem Setup können anstehende Änderungen im aktuellen Projekt einfach committet werden:

```
$ wlc commit
```

1.14 Weblates Python API

1.14.1 Installation

The Python API is shipped separately, you need to install the *Weblate-Client* (wlc) to have it.

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception `wlc.WeblateException`

Base class for all exceptions.

Weblate

class `wlc.Weblate` (*key=""*, *url=None*, *config=None*)

Parameter

- **key** (*str*) – User key
- **url** (*str*) – API server URL, if not specified default is used
- **config** (`wlc.config.WeblateConfig`) – Configuration object, overrides any other parameters.

Access class to the API, define API key and optionally API URL.

get (*path*)

Parameter

path (*str*) – Request path

Rückgabetyyp

object

Performs a single API GET call.

post (*path*, ***kwargs*)

Parameter

path (*str*) – Request path

Rückgabetyyp

object

Performs a single API GET call.

1.14.3 `wlc.config`

`WeblateConfig`

class `wlc.config.WeblateConfig` (*section*='wlc')

Parameter

section (*str*) – Configuration section to use

Configuration file parser following XDG specification.

load (*path*=None)

Parameter

path (*str*) – Path from which to load configuration.

Loads configuration from a file, if none is specified, it loads from the *wlc* configuration file (`~/.config/wlc`) placed in your XDG configuration path (`/etc/xdg/wlc`).

1.14.4 `wlc.main`

`wlc.main.main` (*settings*=None, *stdout*=None, *args*=None)

Parameter

- **settings** (*list*) – Settings to override as list of tuples
- **stdout** (*object*) – stdout file object for printing output, uses `sys.stdout` as default
- **args** (*list*) – Befehlszeilenargumente, die verarbeitet werden sollen, verwendet standardmäßig `sys.args`

Main entry point for command-line interface.

`@wlc.main.register_command` (*command*)

Decorator zur Registrierung der Klasse *Command* im Hauptparser, der von *main()* verwendet wird.

Command

class `wlc.main.Command` (*args*, *config*, *stdout*=None)

Main class for invoking commands.

2.1 Konfigurationsanweisungen

2.1.1 Weblate installieren

Installation über Docker

Mit der Bereitstellung von Weblate per Docker können Sie Ihre persönliche Weblate-Instanz in Sekundenschnelle zum Laufen bringen. Alle Abhängigkeiten von Weblate sind bereits enthalten. PostgreSQL ist als Standarddatenbank eingerichtet.

Hardwareanforderungen

Weblate sollte auf jeder zeitgemäßen Hardware problemlos laufen. Nachfolgend finden Sie die minimale Konfiguration, die erforderlich ist, um Weblate auf einem einzelnen Host zu betreiben (Weblate, Datenbank und Webserver):

- 3 GB Arbeitsspeicher
- 2 CPU-Kerne
- 1 GB Speicherplatz

Je mehr Speicher, desto besser – er wird für das Caching auf allen Ebenen (Dateisystem, Datenbank und Weblate) verwendet.

Viele gleichzeitige Benutzer erhöhen die Anzahl der benötigten CPU-Kerne. Für Hunderte von Übersetzungskomponenten werden mindestens 4 GB RAM empfohlen.

Dies hat schwerwiegende Auswirkungen auf die Leistung der Weboberfläche und beeinträchtigt Funktionen, die von regelmäßigen Auslösern abhängen (z. B. das Übertragen ausstehender Änderungen, Digest-Benachrichtigungen oder Sicherungen).

Bemerkung: Die tatsächlichen Anforderungen an Ihre Weblate-Installation hängen stark von der Größe der darin verwalteten Übersetzungen ab.

Installation

Die folgenden Beispiele gehen davon aus, dass Sie eine funktionierende Docker-Umgebung haben, in der `docker-compose` installiert ist. Anweisungen hierzu finden Sie in der Docker-Dokumentation.

1. Klonen Sie das Weblate-Docker-Repository:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Erstellen Sie eine Datei `docker-compose.override.yml` mit Ihren Einstellungen. Siehe [Docker-Umgebungsvariablen](#) für eine vollständige Liste der Umgebungsvariablen.

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SERVER_EMAIL: weblate@example.com
      WEBLATE_DEFAULT_FROM_EMAIL: weblate@example.com
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for the admin user
      WEBLATE_ADMIN_EMAIL: weblate.admin@example.com
```

Bemerkung: Wenn `WEBLATE_ADMIN_PASSWORD` nicht gesetzt ist, wird der Benutzer admin mit einem zufälligen Passwort erstellt, das beim ersten Start angezeigt wird.

Das mitgelieferte Beispiel lässt Weblate auf Port 80 lauschen. Bearbeiten Sie die Portzuordnung in der Datei `docker-compose.override.yml`, um sie zu ändern.

3. Weblate-Container starten:

```
docker-compose up
```

Viel Spaß beim Einsatz von Weblate, es ist über Port 80 des Containers `weblate` erreichbar.

Geändert in Version 2.15-2: Die Einrichtung hat sich kürzlich geändert, vorher gab es einen separaten Webserver-Container, seit 2.15-2 ist der Webserver in den Weblate-Container eingebettet.

Geändert in Version 3.7.1-6: Im Juli 2019 (beginnend mit dem Tag 3.7.1-6) werden die Container nicht als Root-Benutzer ausgeführt. Dadurch hat sich der exponierte Port von 80 auf 8080 geändert.

Siehe auch:

[Invoking management commands](#)

Auswählen des Docker-Hub-Tags

Sie können die folgenden Tags auf Docker Hub verwenden. Eine vollständige Liste der verfügbaren Tags finden Sie unter <https://hub.docker.com/r/weblate/weblate/tags/>.

Tag-Name	Beschreibung	Anwendungsfall
latest	Stabile Version von Weblate, entspricht der neuesten getagten Version	Fortlaufende Updates in einer Produktionsumgebung
<VERSION>-<F	Stabile Weblate-Version	Gut definierter Einsatz in einer Produktionsumgebung
edge	Stabile Weblate-Version mit Entwicklungsänderungen im Docker-Container (z. B. aktualisierte Abhängigkeiten)	Fortlaufende Updates in einer Staging-Umgebung
edge-<DATE>-	Stabile Weblate-Version mit Entwicklungsänderungen im Docker-Container (z. B. aktualisierte Abhängigkeiten)	Gut definierter Einsatz in einer Staging-Umgebung
bleeding	Weblate-Entwicklungsversion von Git	Fortlaufende Updates zum Testen kommender Weblate-Funktionen
bleeding-<D	Weblate-Entwicklungsversion von Git	Gut definierter Einsatz zum Testen kommender Weblate-Funktionen

Jedes Image wird vor der Veröffentlichung von unserer CI getestet, sodass selbst die *bleeding*-Version sicher verwendbar ist.

Docker-Container mit HTTPS-Unterstützung

Bitte lesen Sie [Installation](#) für allgemeine Bereitstellungsanweisungen, dieser Abschnitt erwähnt nur Unterschiede im Vergleich dazu.

Verwendung eigener SSL-Zertifikate

Neu in Version 3.8-3.

Wenn Sie ein eigenes SSL-Zertifikat haben, das Sie verwenden möchten, legen Sie die Dateien einfach in das Weblate-Daten-Volume (siehe [Docker-Container-Volumes](#)):

- `ssl/fullchain.pem`, das das Zertifikat enthält, einschließlich aller erforderlichen CA-Zertifikate
- `ssl/privkey.pem` mit dem privaten Schlüssel

Beide Dateien müssen demselben Benutzer gehören wie demjenigen, der den Docker-Container startet und die Dateimaske muss auf 600 gesetzt sein (nur lesbar und schreibbar für den besitzenden Benutzer).

Außerdem akzeptiert der Weblate-Container jetzt SSL-Verbindungen auf Port 4443. Sie müssen die Port-Weiterleitung für HTTPS in den Docker Compose Override aufnehmen:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

Wenn Sie bereits andere Sites auf demselben Server hosten, werden die Ports 80 und 443 wahrscheinlich von einem Reverse-Proxy wie NGINX verwendet. Um die HTTPS-Verbindung von NGINX an den Docker-Container zu übergeben, können Sie die folgende Konfiguration verwenden:

```
server {
    listen 443;
    listen [::]:443;

    server_name <SITE_URL>;
    ssl_certificate /etc/letsencrypt/live/<SITE>/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/<SITE>/privkey.pem;

    location / {
        proxy_set_header HOST $host;
        proxy_set_header X-Forwarded-Proto https;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Host $server_name;
        proxy_pass https://127.0.0.1:<EXPOSED_DOCKER_PORT>;
    }
}
```

Ersetzen Sie `<SITE_URL>`, `<SITE>` und `<EXPOSED_DOCKER_PORT>` durch tatsächliche Werte aus Ihrer Umgebung.

Automatische SSL-Zertifikate mit Let's Encrypt

Für den Fall, dass Sie automatisch generierte Let's Encrypt SSL-Zertifikate auf der öffentlichen Installation verwenden möchten, müssen Sie einen Reverse-HTTPS-Proxy in einem zusätzlichen Docker-Container hinzufügen, [https-portal](https://portal) wird dafür verwendet. Dieser wird in der Datei `docker-compose-https.yml` verwendet. Anschließend erstellen Sie eine Datei `docker-compose-https.override.yml` mit Ihren Einstellungen:

```
version: '3'
services:
  weblate:
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for admin user
  https-portal:
    environment:
      DOMAINS: 'weblate.example.com -> http://weblate:8080'
```

Bei jedem Aufruf von **docker-compose** müssen Sie beide Dateien übergeben und dann Folgendes tun:

```
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml ↵
↵build
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml up
```

Aktualisieren des Docker-Containers

Normalerweise ist es eine gute Idee, nur den Weblate-Container zu aktualisieren und den PostgreSQL-Container auf der vorhandenen Version zu belassen, da ein Upgrade von PostgreSQL ziemlich mühsam ist und in den meisten Fällen nicht viel bringt.

Geändert in Version 4.10-1: Seit Weblate 4.10-1 verwendet der Docker-Container Django 4.0, was PostgreSQL 10 oder neuer erfordert. Bitte aktualisieren Sie es vor dem Upgrade von Weblate. Siehe [Upgrade von 4.0 auf 4.10](#) und [Aktualisieren des PostgreSQL-Containers](#).

Sie können dies tun, indem Sie das bestehende Docker-Compose beibehalten und einfach die neuesten Images ziehen und dann neu starten:

```
# Fetch latest versions of the images
docker-compose pull
# Stop and destroy the containers
docker-compose down
# Spawn new containers in the background
docker-compose up -d
# Follow the logs during upgrade
docker-compose logs -f
```

Die Weblate-Datenbank sollte beim ersten Start automatisch migriert werden, und es sollten keine weiteren manuellen Maßnahmen erforderlich sein.

Bemerkung: Upgrades über Hauptversionen hinweg werden von Weblate nicht unterstützt. Wenn Sie zum Beispiel mit der 3.x-Serie arbeiten und auf 4.x aktualisieren möchten, aktualisieren Sie zunächst auf das neueste 4.0.x-y-Image (zum Zeitpunkt der Erstellung dieses Artikels ist es das 4.0.4-5), das die Migration durchführt, und fahren Sie dann mit dem Upgrade auf neuere Versionen fort.

Sie können auch das `docker-compose` Repository aktualisieren, obwohl dies in den meisten Fällen nicht notwendig ist. Siehe [Aktualisieren des PostgreSQL-Containers](#) für die Aktualisierung des PostgreSQL-Servers.

Aktualisieren des PostgreSQL-Containers

PostgreSQL-Container unterstützen kein automatisches Upgrade zwischen Versionen, Sie müssen das Upgrade manuell durchführen. Die folgenden Schritte zeigen eine der Möglichkeiten des Upgrades.

Siehe auch:

<https://github.com/docker-library/postgres/issues/37>

1. Weblate-Container stoppen:

```
docker-compose stop weblate cache
```

2. Datenbank sichern:

```
docker-compose exec database pg_dumpall --clean --username weblate > backup.sql
```

3. Datenbank-Container sperren:

```
docker-compose stop database
```

4. Entfernen des PostgreSQL-Volumes:

```
docker-compose rm -v database
docker volume remove weblate-docker_postgres-data
```

5. Passen Sie `docker-compose.yml` an, um die neue PostgreSQL-Version zu verwenden.

6. Datenbank-Container öffnen:

```
docker-compose up -d database
```

7. Datenbank aus Sicherung wiederherstellen:

```
cat backup.sql | docker-compose exec -T database psql --username weblate --
↳ dbname postgres
```

8. (Optional) Aktualisieren Sie das Passwort für den Weblate-Benutzer. Dies kann bei der Migration auf PostgreSQL 14 oder 15 erforderlich sein, da die Art der Speicherung von Passwörtern geändert wurde:

```
docker-compose exec -T database psql --username weblate --dbname postgres -c
↳ "ALTER USER weblate WITH PASSWORD 'weblate'"
```

9. Alle verbleibenden Container öffnen:

```
docker-compose up -d
```

Administratoranmeldung

Nach der Container-Einrichtung können Sie sich als Benutzer *admin* mit dem in `WEBLATE_ADMIN_PASSWORD` bereitgestellten Passwort oder, falls es nicht festgelegt wurde, mit einem beim ersten Öffnen erzeugten Zufallspasswort anmelden.

Um das Passwort für *admin* zurückzusetzen, öffnen Sie den Container mit dem in `WEBLATE_ADMIN_PASSWORD` neu festgelegten Passwort nochmals.

Siehe auch:

`WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

Anzahl der Prozesse und Speicherverbrauch

Die Anzahl der Mitarbeitervorgänge wird sowohl für uWSGI als auch Celery automatisch auf Grundlage der Anzahl der CPUs bestimmt. Dies funktioniert für die meisten virtuellen Maschinen in der Cloud gut, da sie typischerweise wenig CPUs und große Speicherkapazitäten besitzen.

Für den Fall, dass Sie sehr viele CPU-Kerne haben und auf Speicherprobleme stoßen, versuchen Sie die Zahl der Arbeitskräfte zu reduzieren:

```
environment:
  WEBLATE_WORKERS: 2
```

Sie können auch individuelle Arbeitskräftekategorien feinabstimmen:

```
environment:
  WEB_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Siehe auch:

`WEBLATE_WORKERS`, `CELERY_MAIN_OPTIONS`, `CELERY_NOTIFY_OPTIONS`, `CELERY_MEMORY_OPTIONS`, `CELERY_TRANSLATE_OPTIONS`, `CELERY_BACKUP_OPTIONS`, `CELERY_BEAT_OPTIONS`, `UWSGI_WORKERS`

Horizontale Skalierung

Neu in Version 4.6.

Sie können mehrere Weblate-Container ausführen, um den Dienst horizontal zu skalieren. Das Volume `/app/data` muss von allen Containern gemeinsam genutzt werden, es wird empfohlen, dafür ein Cluster-Dateisystem wie z. B. GlusterFS zu verwenden. Das Volume `/app/cache` sollte für jeden Container separat sein.

Jeder Weblate-Container hat eine definierte Rolle mit der Umgebungsvariablen `WEBLATE_SERVICE`. Bitte folgen Sie sorgfältig der Dokumentation, da einige der Dienste nur einmal im Cluster laufen sollen und auch die Reihenfolge der Dienste wichtig ist.

Ein Beispiel-Setup finden Sie im `docker-compose`-Repo als `docker-compose-split.yml`.

Docker-Umgebungsvariablen

Vieles von der Weblate-*Konfiguration* kann mit den unten beschriebenen Umgebungsvariablen im Docker-Container eingestellt werden.

Wenn Sie eine Einstellung definieren müssen, die nicht über Docker-Umgebungsvariablen zugänglich ist, siehe *Konfiguration über Umgebungsvariablen hinaus*.

Allgemeine Einstellungen

WEBLATE_DEBUG

Konfiguriert den Debugmodus von Django mit *DEBUG*.

Beispiel:

```
environment:
  WEBLATE_DEBUG: 1
```

Siehe auch:

Debugmodus deaktivieren

WEBLATE_LOGLEVEL

Konfiguriert die Ausführlichkeit der Protokollierung.

WEBLATE_LOGLEVEL_DATABASE

Konfiguriert die Protokollierung der Ausführlichkeit der Datenbankabfragen.

WEBLATE_SITE_TITLE

Ändert den Seitentitel, der in der Kopfzeile aller Seiten angezeigt wird.

WEBLATE_SITE_DOMAIN

Konfiguriert die Seitendomain. Dieser Parameter ist erforderlich.

Siehe auch:

Seitendomain richtig einstellen, SITE_DOMAIN

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Legt den Namen und die E-Mail des Website-Administrators fest. Es wird sowohl für die *ADMINS*-Einstellung als auch für die Erstellung des *admin*-Benutzers verwendet (siehe *WEBLATE_ADMIN_PASSWORD* für weitere Informationen dazu).

Beispiel:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

Siehe auch:

Administratoranmeldung, Administratoren richtig konfigurieren, ADMINS

WEBLATE_ADMIN_PASSWORD

Setzt das Passwort für den Benutzer *admin*.

- Wenn nicht gesetzt und der Benutzer *admin* nicht existiert, wird er mit einem zufälligen Passwort erstellt, das beim ersten Start des Containers angezeigt wird.
- Wenn nicht gesetzt und der Benutzer *admin* existiert, wird keine Aktion durchgeführt.

- Wenn gesetzt, wird der *admin*-Benutzer bei jedem Start des Containers angepasst, um `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME` und `WEBLATE_ADMIN_EMAIL` zu entsprechen.

Warnung: Es könnte ein Sicherheitsrisiko darstellen, das Passwort in der Konfigurationsdatei zu speichern. Erwägen Sie, diese Variable nur für die Ersteinrichtung zu verwenden (oder lassen Sie Weblate beim ersten Start ein zufälliges Passwort generieren) oder für die Wiederherstellung des Passworts.

Siehe auch:

Administratoranmeldung, `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_PASSWORD_FILE`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

WEBLATE_ADMIN_PASSWORD_FILE

Legt den Pfad zu einer Datei fest, die das Passwort für den Benutzer *Administrator* enthält.

Siehe auch:

`WEBLATE_ADMIN_PASSWORD`

WEBLATE_SERVER_EMAIL

Die E-Mail-Adresse, von der aus Fehlermeldungen gesendet werden.

Siehe auch:

`SERVER_EMAIL`, *E-Mail-Versand konfigurieren*

WEBLATE_DEFAULT_FROM_EMAIL

Legt die Adresse für ausgehende E-Mails fest.

Siehe auch:

`DEFAULT_FROM_EMAIL`, *E-Mail-Versand konfigurieren*

WEBLATE_CONTACT_FORM

Konfiguriert das Verhalten des Kontaktformulars, siehe *CONTACT_FORM*.

WEBLATE_ALLOWED_HOSTS

Konfiguriert erlaubte HTTP-Hostnamen mit `ALLOWED_HOSTS`.

Der Standardwert ist `*`, der alle Hostnamen zulässt.

Beispiel:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com,example.com
```

Siehe auch:

`ALLOWED_HOSTS`, *Einrichtung zulässiger Hosts*, *Seitendomain richtig einstellen*

WEBLATE_REGISTRATION_OPEN

Legt fest, ob Registrierungen offen sind, durch Umschaltung von `REGISTRATION_OPEN`.

Beispiel:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Konfigurieren Sie über `REGISTRATION_ALLOW_BACKENDS`, welche Authentifizierungsmethoden zum Erstellen eines neuen Kontos verwendet werden können.

Beispiel:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
  ↪oauth2
```

WEBLATE_REGISTRATION_REBIND

Neu in Version 4.16.

Konfiguriert `REGISTRATION_REBIND`.

WEBLATE_TIME_ZONE

Konfiguriert die verwendete Zeitzone in Weblate, siehe `TIME_ZONE`.

Bemerkung: Um die Zeitzone des Docker-Containers selbst zu ändern, verwenden Sie die Umgebungsvariable `TZ`.

Beispiel:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Lässt Weblate davon ausgehen, dass es hinter einem Reverse-HTTPS-Proxy betrieben wird, und bringt Weblate dazu, HTTPS in E-Mails und API-Links zu verwenden oder sichere Markierungen für Cookies zu setzen.

Hinweis: Bitte lesen Sie die `ENABLE_HTTPS`-Dokumentation für mögliche Vorsichtsmaßnahmen.

Bemerkung: Dies führt nicht dazu, dass der Weblate-Container HTTPS-Verbindungen akzeptiert. Sie müssen dies ebenfalls konfigurieren, siehe *Docker-Container mit HTTPS-Unterstützung* für Beispiele.

Beispiel:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Siehe auch:

`ENABLE_HTTPS` Seitendomain richtig einstellen, `WEBLATE_SECURE_PROXY_SSL_HEADER`

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

Neu in Version 4.12.1.

Lässt Weblate das Feld `meta[name=monetization]` im Kopf des Dokuments setzen. Wenn mehrere angegeben sind, wird eines zufällig ausgewählt.

Siehe auch:

`INTERLEDGER_PAYMENT_POINTERS`

WEBLATE_IP_PROXY_HEADER

Ermöglicht es Weblate, die IP-Adresse aus einem beliebigen HTTP-Header zu holen. Verwenden Sie dies, wenn Sie einen Reverse-Proxy vor dem Weblate-Container einsetzen.

Aktiviert `IP_BEHIND_REVERSE_PROXY` und setzt `IP_PROXY_HEADER`.

Bemerkung: Das Format muss mit den Erwartungen von Django übereinstimmen. Django wandelt rohe HTTP-Header-Namen wie folgt um:

- wandelt alle Zeichen in Großbuchstaben um
- ersetzt alle Bindestriche durch Unterstriche
- stellt den ``HTTP_`` Präfix voran

So würde X-Forwarded-For auf HTTP_X_FORWARDED_FOR abgebildet werden.

Beispiel:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Ein Tupel, das eine HTTP-Header/Wert-Kombination darstellt, die bedeutet, dass eine Anfrage sicher ist. Dies wird benötigt, wenn Weblate hinter einem Reverse-Proxy läuft, der die SSL-Terminierung durchführt und keine Standard-HTTPS-Header weitergibt.

Beispiel:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

Siehe auch:

[SECURE_PROXY_SSL_HEADER](#)

WEBLATE_REQUIRE_LOGIN

Aktiviert [REQUIRE_LOGIN](#), um die Authentifizierung auf dem gesamten Weblate zu erzwingen.

Beispiel:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Fügt URL-Ausnahmen für die Authentifizierung hinzu, die für die gesamte Weblate-Installation durch [LOGIN_REQUIRED_URLS_EXCEPTIONS](#) erforderlich sind.

Sie können entweder ganze Einstellungen ersetzen oder den Standardwert mittels der Variablen ADD und REMOVE ändern.

WEBLATE_GOOGLE_ANALYTICS_ID

Konfiguriert die ID für Google Analytics durch Ändern von [GOOGLE_ANALYTICS_ID](#).

WEBLATE_GITHUB_USERNAME

WEBLATE_GITHUB_TOKEN

WEBLATE_GITHUB_HOST

Konfiguriert die Integration von GitHub-Pull-Requests durch Ändern von [GITHUB_CREDENTIALS](#).

Siehe auch:

[GitHub-Pull-Requests](#)

WEBLATE_GITLAB_USERNAME

WEBLATE_GITLAB_TOKEN

WEBLATE_GITLAB_HOST

Konfiguriert die Integration von GitLab-Merge-Requests durch Ändern von *GITLAB_CREDENTIALS*.

Siehe auch:

GitLab-Merge-Requests

WEBLATE_GITEA_USERNAME**WEBLATE_GITEA_TOKEN****WEBLATE_GITEA_HOST**

Konfiguriert die Integration von Gitea-Pull-Requests durch Ändern von *GITEA_CREDENTIALS*.

Siehe auch:

Gitea-Pull-Requests

WEBLATE_PAGURE_USERNAME**WEBLATE_PAGURE_TOKEN****WEBLATE_PAGURE_HOST**

Konfiguriert die Integration von Pagure-Merge-Requests durch Ändern von *PAGURE_CREDENTIALS*.

Siehe auch:

Pagure-Merge-Requests

WEBLATE_BITBUCKETSERVER_USERNAME**WEBLATE_BITBUCKETSERVER_TOKEN****WEBLATE_BITBUCKETSERVER_HOST**

Konfiguriert die Integration von Bitbucket-Server-Pull-Requests durch Ändern von *BITBUCKETSERVER_CREDENTIALS*.

Siehe auch:

Bitbucket Server pull requests

WEBLATE_DEFAULT_PULL_MESSAGE

Konfiguriert den Standardtitel und die Nachricht für Pull Requests über die API durch Ändern von *DEFAULT_PULL_MESSAGE*.

Siehe auch:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

Konfiguriert die Richtlinie zur Sprachvereinfachung, siehe `:setting:'SIMPLIFY_LANGUAGES'`.

WEBLATE_DEFAULT_ACCESS_CONTROL

Konfiguriert die standardmäßige *Zugriffssteuerung* für neue Projekte, siehe *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Legt den Standardwert für *Eingeschränkter Zugriff* für neue Komponenten fest, siehe *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Legt den Standardwert für *Verbreitung von Übersetzungen erlauben* für neue Komponenten fest, siehe *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Konfiguriert *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Konfiguriert *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

Konfiguriert *DEFAULT_SHARED_TM*.

WEBLATE_AKISMET_API_KEY

Konfiguriert den Akismet-API-Schlüssel, siehe *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Konfiguriert die GPG-Signierung von Commits, siehe *WEBLATE_GPG_IDENTITY*.

Siehe auch:

Signing Git commits with GnuPG

WEBLATE_URL_PREFIX

Konfiguriert den URL-Präfix, unter dem Weblate läuft, siehe *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Konfiguriert Prüfungen, die nicht angezeigt werden sollen, siehe *SILENCED_SYSTEM_CHECKS*.

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Ermöglicht die Anpassung des Content-Security-Policy HTTP-Headers.

Siehe auch:

Richtlinie zur Inhaltssicherheit, *CSP_SCRIPT_SRC*, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*,
CSP_STYLE_SRC, *CSP_FONT_SRC*

WEBLATE_LICENSE_FILTER

Konfiguriert *LICENSE_FILTER*.

WEBLATE_LICENSE_REQUIRED

Konfiguriert *LICENSE_REQUIRED*

WEBLATE_WEBSITE_REQUIRED

Konfiguriert *WEBSITE_REQUIRED*

WEBLATE_HIDE_VERSION

Konfiguriert *HIDE_VERSION*.

WEBLATE_BASIC_LANGUAGES

Konfiguriert *BASIC_LANGUAGES*.

WEBLATE_DEFAULT_AUTO_WATCH

Konfiguriert *DEFAULT_AUTO_WATCH*.

WEBLATE_RATELIMIT_ATTEMPTS

WEBLATE_RATELIMIT_LOCKOUT

WEBLATE_RATELIMIT_WINDOW

Neu in Version 4.6.

Konfiguriert den Ratenbegrenzer.

Hinweis: Sie können die Konfiguration für beliebige Ratenbegrenzer-Bereiche festlegen. Fügen Sie dazu das Präfix `WEBLATE_` zu einer der in *Ratenbegrenzung* beschriebenen Einstellungen hinzu.

Siehe auch:

Ratenbegrenzung, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

WEBLATE_API_RATELIMIT_ANON**WEBLATE_API_RATELIMIT_USER**

Neu in Version 4.11.

Konfiguriert die API-Ratenbegrenzung. Der Standardwert ist 100/day für anonyme und 5000/hour für authentifizierte Benutzer.

Siehe auch:

API-Ratenbegrenzung

WEBLATE_ENABLE_HOOKS

Neu in Version 4.13.

Konfiguriert *ENABLE_HOOKS*.

WEBLATE_ENABLE_AVATARS

Neu in Version 4.6.1.

Konfiguriert *ENABLE_AVATARS*.

WEBLATE_AVATAR_URL_PREFIX

Neu in Version 4.15.

Konfiguriert *AVATAR_URL_PREFIX*.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Neu in Version 4.9.

Konfiguriert *LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH*.

WEBLATE_SSH_EXTRA_ARGS

Neu in Version 4.9.

Konfiguriert *SSH_EXTRA_ARGS*.

WEBLATE_BORG_EXTRA_ARGS

Neu in Version 4.9.

Konfiguriert *BORG_EXTRA_ARGS*.

WEBLATE_ENABLE_SHARING

Neu in Version 4.14.1.

Konfiguriert *ENABLE_SHARING*.

WEBLATE_EXTRA_HTML_HEAD

Neu in Version 4.15.

Konfiguriert *EXTRA_HTML_HEAD*.

WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE

Neu in Version 4.15.

Konfiguriert `PRIVATE_COMMIT_EMAIL_TEMPLATE`.

WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN

Neu in Version 4.15.

Konfiguriert `PRIVATE_COMMIT_EMAIL_OPT_IN`.

WEBLATE_CORS_ALLOWED_ORIGINS

Neu in Version 4.16.

Allow CORS requests from given origins.

Beispiel:

```
environment:
  WEBLATE_CORS_ALLOWED_ORIGINS: https://example.com,https://weblate.org
```

Automatische Vorschlagseinstellungen

Geändert in Version 4.13: Automatische Vorschlagsdienste werden jetzt in der Benutzeroberfläche konfiguriert, siehe *Automatische Vorschläge konfigurieren*.

Die vorhandenen Umgebungsvariablen werden bei der Migration auf Weblate 4.13 importiert, sie zu ändern hat jedoch keine weiteren Auswirkungen.

Authentifizierungseinstellungen

LDAP

WEBLATE_AUTH_LDAP_SERVER_URI

WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE

WEBLATE_AUTH_LDAP_USER_ATTR_MAP

WEBLATE_AUTH_LDAP_BIND_DN

WEBLATE_AUTH_LDAP_BIND_PASSWORD

WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE

Pfad zu der Datei, die das LDAP-Server-Verbindungskennwort enthält.

Siehe auch:

`WEBLATE_AUTH_LDAP_BIND_PASSWORD`

WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS

WEBLATE_AUTH_LDAP_USER_SEARCH

WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION

WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER

Konfiguration der LDAP-Authentifizierung.

Beispiel für direkte Bindung:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE: uid=%(user)s,ou=People,dc=example,dc=net
  # map weblate 'full_name' to ldap 'name' and weblate 'email' attribute to
  → 'mail' ldap attribute.
  # another example that can be used with OpenLDAP: 'full_name:cn,email:mail'
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
```

Beispiel für Suche und Bindung:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
```

Beispiel für Vereinigungssuche und Bindung:

```
environment:
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH_UNION: ou=users,dc=example,
  → dc=com|ou=otherusers,dc=example,dc=com
```

Beispiel mit Suche und Bindung gegen Active Directory:

```
environment:
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS: 0
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER: (sAMAccountName=%(user)s)
```

Siehe auch:

LDAP-Authentifizierung

GitHub

WEBLATE_SOCIAL_AUTH_GITHUB_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID

Aktiviert *GitHub-Authentifizierung*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Aktiviert *Bitbucket-Authentifizierung*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Aktiviert *Facebook OAuth 2*.

Google

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS

Aktiviert *Google OAuth 2*.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

Aktiviert *GitLab OAuth 2*.

Gitea

WEBLATE_SOCIAL_AUTH_GITEA_API_URL

WEBLATE_SOCIAL_AUTH_GITEA_KEY

WEBLATE_SOCIAL_AUTH_GITEA_SECRET

Aktiviert die Gitea-Authentifizierung.

Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET

Aktiviert die Azure Active Directory-Authentifizierung, siehe *Microsoft Azure Active Directory*.

Azure Active Directory mit Mandantenunterstützung

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID

Aktiviert die Azure Active Directory-Authentifizierung mit Mandantenunterstützung, siehe :ref:'azure-auth'.

Keycloak

WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET

WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM

WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE

WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE

Aktiviert die Keycloak-Authentifizierung, siehe „Dokumentation“ <https://github.com/python-social-auth/social-core/blob/master/social_core/backends/keycloak.py>‘.

Linux-Anbieter

Sie können die Authentifizierung über die Authentifizierungsdienste von Linux-Anbietern aktivieren, indem Sie die folgenden Variablen auf einen beliebigen Wert setzen.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Aktiviert die Slack-Authentifizierung, siehe [Slack](#).

OpenID Connect

Neu in Version 4.13-1.

WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT

WEBLATE_SOCIAL_AUTH_OIDC_KEY

WEBLATE_SOCIAL_AUTH_OIDC_SECRET

WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY

Konfiguriert die allgemeine OpenID Connect-Integration.

Siehe auch:

[OIDC \(OpenID Connect\)](#)

SAML

Selbstsignierte SAML-Schlüssel werden beim ersten Start des Containers automatisch erzeugt. Wenn Sie eigene Schlüssel verwenden möchten, legen Sie das Zertifikat und den privaten Schlüssel in `/app/data/ssl/saml.crt` und `/app/data/ssl/saml.key` ab.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

WEBLATE_SAML_IDP_IMAGE

WEBLATE_SAML_IDP_TITLE

SAML-Identitätsanbieterereinstellungen, siehe `:ref:'saml-auth'`.

Weitere Authentifizierungseinstellungen

WEBLATE_NO_EMAIL_AUTH

Deaktiviert die E-Mail-Authentifizierung, wenn auf einen beliebigen Wert gesetzt. Siehe [Passwort-Authentifizierung deaktivieren](#).

Einrichtung der PostgreSQL-Datenbank

Die Datenbank wird von `docker-compose.yml` erstellt, daher betreffen diese Einstellungen sowohl Weblate- als auch PostgreSQL-Container.

Siehe auch:

[Datenbankeinrichtung für Weblate](#)

POSTGRES_PASSWORD

PostgreSQL-Passwort.

POSTGRES_PASSWORD_FILE

Pfad zu der Datei, die das PostgreSQL-Passwort enthält. Verwendung als Alternative zu `POSTGRES_PASSWORD`.

POSTGRES_USER

PostgreSQL-Benutzername.

POSTGRES_DATABASE

PostgreSQL-Datenbankname.

POSTGRES_HOST

Hostname oder IP-Adresse des PostgreSQL-Servers. Der Standardwert ist `database`.

POSTGRES_PORT

PostgreSQL-Serverport. Der Standardwert ist `none` (verwendet den Standardwert).

POSTGRES_SSL_MODE

Konfigurieren Sie, wie PostgreSQL SSL bei der Verbindung zum Server handhabt, für mögliche Auswahlen siehe [SSL Mode Descriptions](#)

POSTGRES_ALTER_ROLE

Konfiguriert den Namen der Rolle, der bei Migrationen geändert werden soll, siehe [Weblate für die Verwendung von PostgreSQL konfigurieren](#).

POSTGRES_CONN_MAX_AGE

Neu in Version 4.8.1.

Die Lebensdauer einer Datenbankverbindung als ganze Zahl von Sekunden. Verwenden Sie 0, um Datenbankverbindungen am Ende jeder Anfrage zu schließen (dies ist das Standardverhalten).

Die Aktivierung der Verbindungsaufrechterhaltung führt normalerweise zu mehr offenen Verbindungen zur Datenbank. Bitte passen Sie Ihre Datenbankkonfiguration vor der Aktivierung an.

Beispielkonfiguration:

```
environment:
  POSTGRES_CONN_MAX_AGE: 3600
```

Siehe auch:

`CONN_MAX_AGE`, [Persistent connections](#)

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

Neu in Version 4.9.1.

Deaktiviert serverseitige Cursor in der Datenbank. Dies ist bei einigen **pgbouncer**-Setups notwendig.

Beispielkonfiguration:

```
environment:
  POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Siehe auch:

`DISABLE_SERVER_SIDE_CURSORS`, [Transaction pooling and server-side cursors](#)

Einstellungen für die Datenbanksicherung

Siehe auch:

Gedumpte Daten für Backups

WEBLATE_DATABASE_BACKUP

Konfiguriert den täglichen Datenbank-Dump mit `DATABASE_BACKUP`. Die Voreinstellung ist `plain`.

Einrichtung eines Caching-Servers

Die Verwendung von Redis wird von Weblate dringend empfohlen und Sie müssen eine Redis-Instanz bereitstellen, wenn Sie Weblate in Docker ausführen.

Siehe auch:

Caching einschalten

REDIS_HOST

Der Hostname oder die IP-Adresse des Redis-Servers. Der Standardwert ist `cache`.

REDIS_PORT

Der Port des Redis-Servers. Der Standardwert ist `6379`.

REDIS_DB

Die Nummer der Redis-Datenbank, der Standardwert ist `1`.

REDIS_PASSWORD

Das Redis Server-Passwort, nicht standardmäßig verwendet.

REDIS_PASSWORD_FILE

Pfad zu der Datei, die das Passwort für den Redis-Server enthält.

Siehe auch:

`REDIS_PASSWORD`

REDIS_TLS

Ermöglicht die Verwendung von SSL für Redis-Verbindungen.

REDIS_VERIFY_SSL

Kann verwendet werden, um die SSL-Zertifikatsüberprüfung für Redis-Verbindungen zu deaktivieren.

Einrichtung eines E-Mail-Servers

Damit ausgehende E-Mails funktionieren, müssen Sie einen Mailserver bereitstellen.

Beispiel für eine TLS-Konfiguration:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Beispiel für eine SSL-Konfiguration:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_PORT: 465
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
WEBLATE_EMAIL_USE_TLS: 0
WEBLATE_EMAIL_USE_SSL: 1
```

Siehe auch:*Konfigurieren ausgehender E-Mail***WEBLATE_EMAIL_HOST**

Hostname oder IP-Adresse des Mailservers.

Siehe auch:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_SSL`, `WEBLATE_EMAIL_USE_TLS`,
`EMAIL_HOST`

WEBLATE_EMAIL_PORT

Mailserver-Port, standardmäßig 25.

Siehe auch:

`EMAIL_PORT`

WEBLATE_EMAIL_HOST_USER

Benutzer der E-Mail-Authentifizierung.

Siehe auch:

`EMAIL_HOST_USER`

WEBLATE_EMAIL_HOST_PASSWORD

Passwort für die E-Mail-Authentifizierung.

Siehe auch:

`EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Pfad zu der Datei, die das Kennwort für die E-Mail-Authentifizierung enthält.

Siehe auch:

`WEBLATE_EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_USE_SSL

Ob eine implizite (sichere) TLS-Verbindung für die Kommunikation mit dem SMTP-Server verwendet werden soll. In den meisten E-Mail-Dokumentationen wird diese Art der TLS-Verbindung als SSL bezeichnet. Sie wird im Allgemeinen auf Port 465 eingesetzt. Wenn Sie Probleme haben, sehen Sie sich die explizite TLS-Einstellung `WEBLATE_EMAIL_USE_TLS` an.

Geändert in Version 4.11: Die SSL/TLS-Unterstützung wird automatisch auf der Grundlage des `WEBLATE_EMAIL_PORT` aktiviert.

Siehe auch:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_TLS`, `EMAIL_USE_SSL`

WEBLATE_EMAIL_USE_TLS

Ob eine (sichere) TLS-Verbindung verwendet werden soll, wenn mit dem SMTP-Server kommuniziert wird. Dies wird für explizite TLS-Verbindungen verwendet, im Allgemeinen auf Port 587 oder 25. Wenn Sie feststellen, dass Verbindungen hängen bleiben, sehen Sie sich die implizite TLS-Einstellung `WEBLATE_EMAIL_USE_SSL` an.

Geändert in Version 4.11: Die SSL/TLS-Unterstützung wird automatisch auf der Grundlage des `WEBLATE_EMAIL_PORT` aktiviert.

Siehe auch:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_SSL`, `EMAIL_USE_TLS`

WEBLATE_EMAIL_BACKEND

Konfiguriert das Django-Backend, das für den Versand von E-Mails verwendet werden soll.

Siehe auch:

E-Mail-Versand konfigurieren, `EMAIL_BACKEND`

WEBLATE_AUTO_UPDATE

Konfiguriert, ob und wie Weblate Repositorys aktualisieren soll.

Siehe auch:

AUTO_UPDATE

Bemerkung: Dies ist eine boolesche Einstellung (verwenden Sie `"true"` oder `"false"`).

Integration der Website

WEBLATE_GET_HELP_URL

Konfiguriert *GET_HELP_URL*.

WEBLATE_STATUS_URL

Konfiguriert *STATUS_URL*.

WEBLATE_LEGAL_URL

Konfiguriert *LEGAL_URL*.

WEBLATE_PRIVACY_URL

Konfiguriert *PRIVACY_URL*.

Fehlerberichterstattung

Es wird empfohlen, Fehler bei der Installation systematisch zu sammeln, siehe *Sammeln von Fehlerberichten*.

Um die Unterstützung für Rollbar zu aktivieren, stellen Sie Folgendes ein:

ROLLBAR_KEY

Ihr Rollbar-Postserver-Zugriffstoken.

ROLLBAR_ENVIRONMENT

Ihre Rollbar-Umgebung, standardmäßig `production`.

Um die Unterstützung für Sentry zu aktivieren, stellen Sie Folgendes ein:

SENTRY_DSN

Ihr Sentry-DSN.

SENTRY_ENVIRONMENT

Ihre Sentry-Umgebung (optional).

Lokalisierung CDN

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

Neu in Version 4.2.1.

Konfiguration für *JavaScript-Lokalisierung CDN*.

Der `WEBLATE_LOCALIZE_CDN_PATH` ist ein Pfad innerhalb des Containers. Er sollte auf dem dauerhaften Volume und nicht im flüchtigen Speicher gespeichert werden.

Eine der Möglichkeiten ist die Speicherung im Weblate-Datenverzeichnis:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Bemerkung: Sie sind für die Einrichtung der Bereitstellung der von Weblate generierten Dateien verantwortlich, es speichert nur die Dateien am konfigurierten Speicherort.

Siehe auch:

weblate-cdn, `LOCALIZE_CDN_URL`, `LOCALIZE_CDN_PATH`

Ändern aktivierter Apps, Qualitätsprüfungen, Erweiterungen oder Autofixes

Neu in Version 3.8-5.

Die integrierte Konfiguration der aktivierten Qualitätsprüfungen, Erweiterungen oder Autofixes kann durch die folgenden Variablen angepasst werden:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Beispiel:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
  ↳ SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

Siehe auch:

`CHECK_LIST`, `AUTOFIX_LIST`, `WEBLATE_ADDONS`, `INSTALLED_APPS`

Container-Einstellungen

WEBLATE_WORKERS

Neu in Version 4.6.1.

Basisanzahl der im Container laufenden Arbeitsprozesse. Wenn sie nicht festgelegt ist, wird sie automatisch beim Start des Containers anhand der Anzahl der verfügbaren CPU-Kerne ermittelt.

Wird zur Bestimmung von `CELERY_MAIN_OPTIONS`, `CELERY_NOTIFY_OPTIONS`, `CELERY_MEMORY_OPTIONS`, `CELERY_TRANSLATE_OPTIONS`, `CELERY_BACKUP_OPTIONS`, `CELERY_BEAT_OPTIONS` und `WEB_WORKERS` verwendet. Sie können diese Einstellungen zur Feinabstimmung nutzen.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

Mit diesen Variablen können Sie die Celery-Worker-Optionen anpassen. Es kann nützlich sein, die Parallelität anzupassen (`--concurrency 16`) oder eine andere Pool-Implementierung zu verwenden (`--pool=gevent`).

Standardmäßig basiert die Anzahl der gleichzeitigen Worker auf `WEBLATE_WORKERS`.

Beispiel:

```
environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```

Siehe auch:

[Celery-Worker-Optionen](#), [Hintergrundaufgaben mit Celery](#)

WEB_WORKERS

Konfigurieren Sie, wie viele uWSGI-Worker ausgeführt werden sollen.

Der Standardwert ist `WEBLATE_WORKERS`.

Beispiel:

```
environment:
  WEB_WORKERS: 32
```

WEBLATE_SERVICE

Legt fest, welche Dienste innerhalb des Containers ausgeführt werden sollen. Verwenden Sie dies für [Horizontale Skalierung](#).

Folgende Dienste sind definiert:

celery-beat

Celery-Aufgabenplaner, es sollte nur eine Instanz ausgeführt werden. Dieser Container ist auch für die Migrationen der Datenbankstruktur zuständig und sollte vor den anderen gestartet werden.

celery-backup

Celery-Worker für Backups, es sollte nur eine Instanz laufen.

celery-celery

Allgemeiner Celery-Worker.

celery-memory

Übersetzungsspeicher Celery-Worker.

celery-notify

Benachrichtigungen Celery-Worker.

celery-translate

Automatische Übersetzung Celery-Worker.

web

Webserver.

Docker-Container-Volumes

Es gibt zwei Volumes (Daten und Cache), die vom Weblate-Container exportiert werden. Die anderen Service-Container (PostgreSQL oder Redis) verfügen ebenfalls über Datenvolumina, die jedoch in diesem Dokument nicht behandelt werden.

Das Daten-Volume wird verwendet, um dauerhafte Weblate-Daten wie geklonte Repositorys zu speichern oder die Weblate-Installation anzupassen.

Die Platzierung des Docker-Volumes auf dem Host-System hängt von Ihrer Docker-Konfiguration ab, aber normalerweise wird es in `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/` gespeichert (der Pfad besteht aus dem Namen Ihres Docker-Compose-Verzeichnisses, dem Container und den Volume-Namen). Im Container wird es als `/app/data` eingehängt.

Das Cache-Volume wird als `/app/cache` gemountet und dient zum Speichern von statischen Dateien und `CACHE_DIR`. Sein Inhalt wird beim Start des Containers neu erstellt und das Volume kann mit einem ephemeren Dateisystem wie `tmpfs` eingebunden werden.

Wenn Sie die Volumes manuell erstellen, sollten die Verzeichnisse der UID 1000 gehören, da dies der im Container verwendete Benutzer ist.

Siehe auch:

[Dokumentation der Docker-Volumes](#)

Konfiguration über Umgebungsvariablen hinaus

Docker-Umgebungsvariablen sind dazu gedacht, die meisten *Konfigurationseinstellungen*, die für Weblate-Installationen von Bedeutung sind, sichtbar zu machen.

Wenn Sie eine Einstellung finden, die nicht als Umgebungsvariable verfügbar ist und Sie glauben, dass sie es sein sollte, können Sie *fordern, dass sie in einer zukünftigen Version von Weblate* verfügbar gemacht wird.

Wenn Sie eine Einstellung ändern müssen, die nicht als Docker-Umgebungsvariable verfügbar ist, können Sie dies trotzdem tun, entweder *über das Daten-Volume* oder *durch Erweitern des Docker-Images*.

Siehe auch:

[Anpassen von Weblate](#)

Überschreiben von Einstellungen des Daten-Volumes

Sie können eine Datei unter `/app/data/settings-override.py` erstellen, d. h. in der Wurzel des *Daten-Volume*, um Einstellungen zu erweitern oder zu überschreiben, die durch Umgebungsvariablen definiert wurden.

Überschreiben von Einstellungen durch Erweitern des Docker-Images

Um Einstellungen auf der Ebene des Docker-Images statt des Daten-Volumes zu überschreiben:

1. *Erstellen Sie ein benutzerdefiniertes Python-Paket.*
2. Fügen Sie ein Modul zu Ihrem Paket hinzu, das alle Einstellungen von `weblate.settings_docker` importiert.

Innerhalb der Beispiel-Paketstruktur, die unter *Erstellung eines Python-Moduls* definiert ist, könnten Sie zum Beispiel eine Datei unter `weblate_customization/weblate_customization/settings.py` mit dem folgenden Anfangscode erstellen:

```
from weblate.settings_docker import *
```

3. Erstellen Sie ein benutzerdefiniertes Dockerfile, das vom offiziellen Weblate-Docker-Image erbt, dann Ihr Paket installiert und die Umgebungsvariable `DJANGO_SETTINGS_MODULE` auf Ihr Einstellungsmodul verweist:

```
FROM weblate/weblate

USER root

COPY weblate_customization /usr/src/weblate_customization
RUN pip install --no-cache-dir /usr/src/weblate_customization
ENV DJANGO_SETTINGS_MODULE=weblate_customization.settings

USER 1000
```

4. Anstatt das offizielle Weblate-Docker-Image zu verwenden, erstellen Sie ein benutzerdefiniertes Image aus dieser Dockerfile-Datei.

Es gibt *keine saubere Möglichkeit*, dies mit `docker-compose.override.yml` zu tun. Sie *könnten* `build: .` zu dem `weblate` Knoten in dieser Datei hinzufügen, aber dann wird Ihr benutzerdefiniertes Image als `weblate/weblate` in Ihrem System markiert, was problematisch sein könnte.

Anstatt also `docker-compose.yml` direkt unverändert aus dem offiziellen *Repository* zu verwenden und es durch `docker-compose.override.yml` zu erweitern, sollten Sie eine Kopie der offiziellen `docker-compose.yml` Datei erstellen und Ihre Kopie so bearbeiten, dass `image: weblate/weblate` durch `build: .` ersetzt wird.

Weitere Informationen zum Erstellen von Images aus der Quelle bei Verwendung von `docker-compose` finden Sie in der ‚Compose-Datei-Build-Referenz‘.

5. Erweitern Sie Ihr benutzerdefiniertes Einstellungsmodul, um Einstellungen zu definieren oder neu zu definieren.

Sie können Einstellungen vor oder nach der obigen Importanweisung definieren, um festzulegen, welche Einstellungen Vorrang haben. Vor der Importanweisung definierte Einstellungen können durch Umgebungsvariablen und im Data-Volume definierte Einstellungsüberschreibungen außer Kraft gesetzt werden. Einstellungen, die nach der Importanweisung definiert werden, können nicht überschrieben werden.

Sie können auch noch weiter gehen. Zum Beispiel können Sie einige der Dinge reproduzieren, die `weblate.docker_settings` tut, wie z. B. Einstellungen als Umgebungsvariablen offenlegen oder das Überschreiben von Einstellungen aus Python-Dateien im Daten-Volume erlauben.

Ersetzen des Logos und anderer statischer Dateien

Neu in Version 3.8-5.

Die mit Weblate gelieferten statischen Dateien können überschrieben werden, indem sie in `/app/data/python/customize/static` abgelegt werden (siehe [Docker-Container-Volumes](#)). Zum Beispiel wird durch die Erstellung von `/app/data/python/customize/static/favicon.ico` das Favicon ersetzt.

Hinweis: Die Dateien werden beim Start des Containers an den entsprechenden Ort kopiert, so dass ein Neustart von Weblate erforderlich ist, wenn der Inhalt des Volumes geändert wurde.

Dieser Ansatz kann auch verwendet werden, um Weblate-Vorlagen außer Kraft zu setzen. Zum Beispiel können *Rechtliche Grundlagen*-Dokumente in `/app/data/python/customize/templates/legal/documents` platziert werden.

Alternativ können Sie auch ein eigenes Modul einbinden (siehe [Anpassen von Weblate](#)) und es zum Beispiel als separates Volume dem Docker-Container hinzufügen:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/weblate_
      ↪ customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Konfigurieren des PostgreSQL-Servers

Der PostgreSQL-Container verwendet die Standardkonfiguration von PostgreSQL, die die CPU-Kerne und den Speicher nicht effektiv nutzt. Es wird empfohlen, die Konfiguration anzupassen, um die Leistung zu verbessern.

Die Konfiguration kann wie in *Datenbankkonfiguration* unter https://hub.docker.com/_/postgres beschrieben angepasst werden. Die für Ihre Umgebung passende Konfiguration kann mit <https://pgtune.leopard.in.ua/> generiert werden.

Container-Interns

Der Container verwendet **supervisor**, um einzelne Dienste zu starten. Im Falle von *Horizontale Skalierung* wird nur ein einzelner Dienst in einem Container gestartet.

Um den Status der Dienste zu überprüfen, verwenden Sie:

```
docker-compose exec --user weblate weblate supervisorctl status
```

Für jede Celery-Warteschlange gibt es eigene Dienste (siehe [Hintergrundaufgaben mit Celery](#) für Details). Sie können die Verarbeitung einiger Aufgaben stoppen, indem Sie den entsprechenden Worker stoppen:

```
docker-compose exec --user weblate weblate supervisorctl stop celery-translate
```

Installation unter Debian und Ubuntu

Hardwareanforderungen

Weblate sollte auf jeder zeitgemäßen Hardware problemlos laufen. Nachfolgend finden Sie die minimale Konfiguration, die erforderlich ist, um Weblate auf einem einzelnen Host zu betreiben (Weblate, Datenbank und Webserver):

- 3 GB Arbeitsspeicher
- 2 CPU-Kerne
- 1 GB Speicherplatz

Je mehr Speicher, desto besser – er wird für das Caching auf allen Ebenen (Dateisystem, Datenbank und Weblate) verwendet.

Viele gleichzeitige Benutzer erhöhen die Anzahl der benötigten CPU-Kerne. Für Hunderte von Übersetzungskomponenten werden mindestens 4 GB RAM empfohlen.

Dies hat schwerwiegende Auswirkungen auf die Leistung der Weboberfläche und beeinträchtigt Funktionen, die von regelmäßigen Auslösern abhängen (z. B. das Übertragen ausstehender Änderungen, Digest-Benachrichtigungen oder Sicherungen).

Bemerkung: Die tatsächlichen Anforderungen an Ihre Weblate-Installation hängen stark von der Größe der darin verwalteten Übersetzungen ab.

Installation

Systemvoraussetzungen

Installieren Sie die Abhängigkeiten, die für die Erstellung der Python-Module erforderlich sind (siehe [Software-Anforderungen](#)):

```
apt install -y \
    libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-dev \
    libffi-dev libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev \
    libacl1-dev libssl-dev libpq-dev libjpeg-dev build-essential \
    python3-gdbm python3-dev python3-pip python3-virtualenv virtualenv git
```

Installieren Sie die gewünschten optionalen Abhängigkeiten abhängig von den Funktionen, die Sie verwenden wollen (siehe [Optionale Abhängigkeiten](#)):

```
apt install -y \
    tesseract-ocr libtesseract-dev liblptonica-dev \
    libldap2-dev libldap-common libsasl2-dev \
    libxmlsec1-dev
```

Installieren Sie optional Software für den Betrieb des Produktionsservers, siehe [Laufender Server](#), [Datenbankeinrichtung für Weblate](#), [Hintergrundaufgaben mit Celery](#). Je nach Größe Ihrer Installation sollten Sie diese Komponenten auf getrennten Servern laufen lassen.

Lokale Installationsanweisungen:

```
# Web server option 1: NGINX and uWSGI
apt install -y nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
apt install -y apache2 libapache2-mod-wsgi-py3
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# Caching backend: Redis
apt install -y redis-server

# Database server: PostgreSQL
apt install -y postgresql postgresql-contrib

# SMTP server
apt install -y exim4
```

Python-Module

Hinweis: Sie können Einstellungen vor oder nach der obigen Importanweisung definieren, um festzulegen, welche Einstellungen Vorrang haben. Vor der Importanweisung definierte Einstellungen können durch Umgebungsvariablen und im Daten-Volumen definierte Einstellungsüberschreibungen außer Kraft gesetzt werden. Einstellungen, die nach der Importanweisung definiert werden, können nicht überschrieben werden.

1. Erstellen Sie die Virtualenv für Weblate:

```
virtualenv ~/weblate-env
```

2. Aktivieren Sie die Virtualenv für Weblate:

```
. ~/weblate-env/bin/activate
```

3. Installieren Sie Weblate einschließlich aller optionalen Abhängigkeiten:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Bitte prüfen Sie *Optionale Abhängigkeiten* für die Feinabstimmung der optionalen Abhängigkeiten.

Bemerkung: Auf einigen Linux-Distributionen schlägt die Ausführung von Weblate mit einem libffi-Fehler fehl:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Dies wird durch die Inkompatibilität von Binärpaketen verursacht, die über PyPI mit der Distribution verteilt werden. Um dies zu beheben, müssen Sie das Paket auf Ihrem System neu erstellen:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate konfigurieren

Bemerkung: Im Folgenden wird davon ausgegangen, dass die von Weblate verwendete Virtualenv aktiviert ist (durch Ausführung von `./~/weblate-env/bin/activate`). Falls nicht, geben Sie den vollständigen Pfad zum Befehl **weblate** als `~/weblate-env/bin/weblate` an.

1. Kopieren Sie die Datei `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` nach `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Passen Sie die Werte in der neuen Datei `settings.py` nach Ihren Wünschen an. Sie müssen zumindest die Datenbank-Zugangsdaten und den geheimen Django-Schlüssel angeben, aber Sie werden mehr Änderungen für die Produktionseinrichtung benötigen, siehe [Anpassen der Konfiguration](#).
3. Erstellen Sie die Datenbank und ihre Struktur für Weblate (die Beispieleinstellungen verwenden PostgreSQL, sehen Sie sich [Datenbankeinrichtung für Weblate](#) für ein produktionsreifes Setup an):

```
weblate migrate
```

4. Erstellen Sie ein Benutzerkonto für den Administrator und kopieren Sie dessen Passwort in die Zwischenablage, und speichern Sie es zur späteren Verwendung:

```
weblate createadmin
```

5. Sammeln Sie die statischen Dateien für Ihren Webserver (siehe [Laufender Server](#) und [Bereitstellung statischer Dateien](#)):

```
weblate collectstatic
```

6. Komprimieren Sie die JavaScript- und CSS-Dateien (optional, siehe [Client-Assets komprimieren](#)):

```
weblate compress
```

7. Starten Sie die Celery-Arbeiter. Dies ist für Entwicklungszwecke nicht notwendig, wird aber ansonsten dringend empfohlen. [Hintergrundaufgaben mit Celery](#) hat mehr Informationen:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Starten Sie den Entwicklungsserver (:ref:'server' beschreibt ein Produktionssetup):

```
weblate runserver
```

Nach der Installation

Herzlichen Glückwunsch, Ihr Weblate-Server läuft jetzt und Sie können ihn nutzen.

- Sie können jetzt auf Weblate unter `http://localhost:8000/` zugreifen.
- Melden Sie sich mit den bei der Installation erhaltenen Administrator-Zugangsdaten an oder registrieren Sie sich mit neuen Benutzern.
- Sie können jetzt Weblate-Befehle mit dem Befehl **weblate** ausführen, wenn Weblate virtualenv aktiv ist, siehe [Management commands](#).
- Sie können den Testserver mit Strg+C anhalten.
- Überprüfen Sie mögliche Probleme mit Ihrer Installation entweder unter der URL `/manage/performance/` (siehe [Verwaltungsoberfläche](#)) oder mit **weblate check --deploy**, siehe [Produktionseinrichtung](#).

Hinzufügen einer Übersetzung

1. Öffnen Sie die Verwaltungsoberfläche (<http://localhost:8000/create/project/>) und erstellen Sie das Projekt, das Sie übersetzen möchten. Siehe [Projektkonfiguration](#) für weitere Details.

Sie müssen hier nur den Namen des Projekts und dessen Website angeben.

2. Erstellen Sie eine Komponente, die das eigentliche Objekt für die Übersetzung ist – sie verweist auf das VCS-Repository und wählt die zu übersetzenden Dateien aus. Siehe [Komponentenkonfiguration](#) für weitere Details.

Die wichtigen Felder hier, um übersetzbare Dateien zu finden, sind: *Name der Komponente*, *Quellcode-Repository* und *Dateimaske*. Weblate unterstützt eine Vielzahl von Formaten, darunter *GNU gettext*, *Android String-Ressourcen*, *Apple iOS-Zeichenketten*, *Java properties*, *Stringsdict-Format* oder *Fluent-Format*, siehe [Unterstützte Dateiformate](#) für weitere Details.

3. Sobald die Schritte oben abgeschlossen sind (je nach Größe Ihres VCS-Repositorys und der Anzahl der zu übersetzenden Nachrichten kann dies ein langwieriger Prozess sein), können Sie mit dem Übersetzen beginnen.

Installation unter SUSE und openSUSE

Hardwareanforderungen

Weblate sollte auf jeder zeitgemäßen Hardware problemlos laufen. Nachfolgend finden Sie die minimale Konfiguration, die erforderlich ist, um Weblate auf einem einzelnen Host zu betreiben (Weblate, Datenbank und Webserver):

- 3 GB Arbeitsspeicher
- 2 CPU-Kerne
- 1 GB Speicherplatz

Je mehr Speicher, desto besser – er wird für das Caching auf allen Ebenen (Dateisystem, Datenbank und Weblate) verwendet.

Viele gleichzeitige Benutzer erhöhen die Anzahl der benötigten CPU-Kerne. Für Hunderte von Übersetzungskomponenten werden mindestens 4 GB RAM empfohlen.

Dies hat schwerwiegende Auswirkungen auf die Leistung der Weboberfläche und beeinträchtigt Funktionen, die von regelmäßigen Auslösern abhängen (z. B. das Übertragen ausstehender Änderungen, Digest-Benachrichtigungen oder Sicherungen).

Bemerkung: Die tatsächlichen Anforderungen an Ihre Weblate-Installation hängen stark von der Größe der darin verwalteten Übersetzungen ab.

Installation

Systemvoraussetzungen

Installieren Sie die Abhängigkeiten, die für die Erstellung der Python-Module erforderlich sind (siehe [Software-Anforderungen](#)):

```
zypper install \
  libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
  libyaml-devel libffi-devel cairo-devel pango-devel \
  gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
  python3-devel git
```

Installieren Sie die gewünschten optionalen Abhängigkeiten abhängig von den Funktionen, die Sie verwenden wollen (siehe [Optionale Abhängigkeiten](#)):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libsasl2-devel
zypper install libxmlsec1-devel
```

Installieren Sie optional Software für den Betrieb des Produktionsservers, siehe *Laufender Server*, *Datenbankeinrichtung für Weblate*, *Hintergrundaufgaben mit Celery*. Je nach Größe Ihrer Installation sollten Sie diese Komponenten auf getrennten Servern laufen lassen.

Lokale Installationsanweisungen:

```
# Web server option 1: NGINX and uWSGI
zypper install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
zypper install apache2 apache2-mod_wsgi

# Caching backend: Redis
zypper install redis-server

# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Python-Module

Hinweis: Sie können Einstellungen vor oder nach der obigen Importanweisung definieren, um festzulegen, welche Einstellungen Vorrang haben. Vor der Importanweisung definierte Einstellungen können durch Umgebungsvariablen und im Daten-Volumen definierte Einstellungsüberschreibungen außer Kraft gesetzt werden. Einstellungen, die nach der Importanweisung definiert werden, können nicht überschrieben werden.

1. Erstellen Sie die Virtualenv für Weblate:

```
virtualenv ~/weblate-env
```

2. Aktivieren Sie die Virtualenv für Weblate:

```
. ~/weblate-env/bin/activate
```

3. Installieren Sie Weblate einschließlich aller optionalen Abhängigkeiten:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Bitte prüfen Sie *Optionale Abhängigkeiten* für die Feinabstimmung der optionalen Abhängigkeiten.

Bemerkung: Auf einigen Linux-Distributionen schlägt die Ausführung von Weblate mit einem libffi-Fehler fehl:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi_
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Dies wird durch die Inkompatibilität von Binärpaketen verursacht, die über PyPI mit der Distribution verteilt werden. Um dies zu beheben, müssen Sie das Paket auf Ihrem System neu erstellen:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate konfigurieren

Bemerkung: Im Folgenden wird davon ausgegangen, dass die von Weblate verwendete Virtualenv aktiviert ist (durch Ausführung von `. ~/weblate-env/bin/activate`). Falls nicht, geben Sie den vollständigen Pfad zum Befehl **weblate** als `~/weblate-env/bin/weblate` an.

1. Kopieren Sie die Datei `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` nach `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Passen Sie die Werte in der neuen Datei `settings.py` nach Ihren Wünschen an. Sie müssen zumindest die Datenbank-Zugangsdaten und den geheimen Django-Schlüssel angeben, aber Sie werden mehr Änderungen für die Produktionseinrichtung benötigen, siehe [Anpassen der Konfiguration](#).
3. Erstellen Sie die Datenbank und ihre Struktur für Weblate (die Beispieleinstellungen verwenden PostgreSQL, sehen Sie sich [Datenbankeinrichtung für Weblate](#) für ein produktionsreifes Setup an):

```
weblate migrate
```

4. Erstellen Sie ein Benutzerkonto für den Administrator und kopieren Sie dessen Passwort in die Zwischenablage, und speichern Sie es zur späteren Verwendung:

```
weblate createadmin
```

5. Sammeln Sie die statischen Dateien für Ihren Webserver (siehe [Laufender Server](#) und [Bereitstellung statischer Dateien](#)):

```
weblate collectstatic
```

6. Komprimieren Sie die JavaScript- und CSS-Dateien (optional, siehe [Client-Assets komprimieren](#)):

```
weblate compress
```

7. Starten Sie die Celery-Arbeiter. Dies ist für Entwicklungszwecke nicht notwendig, wird aber ansonsten dringend empfohlen. [Hintergrundaufgaben mit Celery](#) hat mehr Informationen:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Starten Sie den Entwicklungsserver (ref:‘server‘ beschreibt ein Produktionssetup):

```
weblate runserver
```

Nach der Installation

Herzlichen Glückwunsch, Ihr Weblate-Server läuft jetzt und Sie können ihn nutzen.

- Sie können jetzt auf Weblate unter `http://localhost:8000/` zugreifen.
- Melden Sie sich mit den bei der Installation erhaltenen Administrator-Zugangsdaten an oder registrieren Sie sich mit neuen Benutzern.
- Sie können jetzt Weblate-Befehle mit dem Befehl **weblate** ausführen, wenn Weblate virtualenv aktiv ist, siehe [Management commands](#).

- Sie können den Testserver mit Strg+C anhalten.
- Überprüfen Sie mögliche Probleme mit Ihrer Installation entweder unter der URL `/manage/performance/` (siehe [Verwaltungsoberfläche](#)) oder mit `weblate check --deploy`, siehe [Produktionseinrichtung](#).

Hinzufügen einer Übersetzung

1. Öffnen Sie die Verwaltungsoberfläche (`http://localhost:8000/create/project/`) und erstellen Sie das Projekt, das Sie übersetzen möchten. Siehe [Projektkonfiguration](#) für weitere Details.

Sie müssen hier nur den Namen des Projekts und dessen Website angeben.

2. Erstellen Sie eine Komponente, die das eigentliche Objekt für die Übersetzung ist – sie verweist auf das VCS-Repository und wählt die zu übersetzenden Dateien aus. Siehe [Komponentenkonfiguration](#) für weitere Details.

Die wichtigen Felder hier, um übersetzbare Dateien zu finden, sind: *Name der Komponente*, *Quellcode-Repository* und *Dateimasken*. Weblate unterstützt eine Vielzahl von Formaten, darunter *GNU gettext*, *Android String-Ressourcen*, *Apple iOS-Zeichenketten*, *Java properties*, *Stringsdict-Format* oder *Fluent-Format*, siehe [Unterstützte Dateiformate](#) für weitere Details.

3. Sobald die Schritte oben abgeschlossen sind (je nach Größe Ihres VCS-Repositorys und der Anzahl der zu übersetzenden Nachrichten kann dies ein langwieriger Prozess sein), können Sie mit dem Übersetzen beginnen.

Installation unter RedHat, Fedora und CentOS

Hardwareanforderungen

Weblate sollte auf jeder zeitgemäßen Hardware problemlos laufen. Nachfolgend finden Sie die minimale Konfiguration, die erforderlich ist, um Weblate auf einem einzelnen Host zu betreiben (Weblate, Datenbank und Webserver):

- 3 GB Arbeitsspeicher
- 2 CPU-Kerne
- 1 GB Speicherplatz

Je mehr Speicher, desto besser – er wird für das Caching auf allen Ebenen (Dateisystem, Datenbank und Weblate) verwendet.

Viele gleichzeitige Benutzer erhöhen die Anzahl der benötigten CPU-Kerne. Für Hunderte von Übersetzungskomponenten werden mindestens 4 GB RAM empfohlen.

Dies hat schwerwiegende Auswirkungen auf die Leistung der Weboberfläche und beeinträchtigt Funktionen, die von regelmäßigen Auslösern abhängen (z. B. das Übertragen ausstehender Änderungen, Digest-Benachrichtigungen oder Sicherungen).

Bemerkung: Die tatsächlichen Anforderungen an Ihre Weblate-Installation hängen stark von der Größe der darin verwalteten Übersetzungen ab.

Installation

Systemvoraussetzungen

Installieren Sie die Abhängigkeiten, die für die Erstellung der Python-Module erforderlich sind (siehe *Software-Anforderungen*):

```
dnf install \
  libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
  libyaml-devel libffi-devel cairo-devel pango-devel \
  gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
  python3-devel git
```

Installieren Sie die gewünschten optionalen Abhängigkeiten abhängig von den Funktionen, die Sie verwenden wollen (siehe *Optionale Abhängigkeiten*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
dnf install libldap2-devel libsasl2-devel
dnf install libxmlsec1-devel
```

Installieren Sie optional Software für den Betrieb des Produktionsservers, siehe *Laufender Server*, *Datenbankeinrichtung für Weblate*, *Hintergrundaufgaben mit Celery*. Je nach Größe Ihrer Installation sollten Sie diese Komponenten auf getrennten Servern laufen lassen.

Lokale Installationsanweisungen:

```
# Web server option 1: NGINX and uWSGI
dnf install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
dnf install apache2 apache2-mod_wsgi

# Caching backend: Redis
dnf install redis

# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Python-Module

Hinweis: Sie können Einstellungen vor oder nach der obigen Importanweisung definieren, um festzulegen, welche Einstellungen Vorrang haben. Vor der Importanweisung definierte Einstellungen können durch Umgebungsvariablen und im Daten-Volume definierte Einstellungsüberschreibungen außer Kraft gesetzt werden. Einstellungen, die nach der Importanweisung definiert werden, können nicht überschrieben werden.

1. Erstellen Sie die Virtualenv für Weblate:

```
virtualenv ~/weblate-env
```

2. Aktivieren Sie die Virtualenv für Weblate:

```
. ~/weblate-env/bin/activate
```

3. Installieren Sie Weblate einschließlich aller optionalen Abhängigkeiten:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Bitte prüfen Sie *Optionale Abhängigkeiten* für die Feinabstimmung der optionalen Abhängigkeiten.

Bemerkung: Auf einigen Linux-Distributionen schlägt die Ausführung von Weblate mit einem libffi-Fehler fehl:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Dies wird durch die Inkompatibilität von Binärpaketen verursacht, die über PyPI mit der Distribution verteilt werden. Um dies zu beheben, müssen Sie das Paket auf Ihrem System neu erstellen:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate konfigurieren

Bemerkung: Im Folgenden wird davon ausgegangen, dass die von Weblate verwendete Virtualenv aktiviert ist (durch Ausführung von `. ~/weblate-env/bin/activate`). Falls nicht, geben Sie den vollständigen Pfad zum Befehl **weblate** als `~/weblate-env/bin/weblate` an.

1. Kopieren Sie die Datei `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` nach `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Passen Sie die Werte in der neuen Datei `settings.py` nach Ihren Wünschen an. Sie müssen zumindest die Datenbank-Zugangsdaten und den geheimen Django-Schlüssel angeben, aber Sie werden mehr Änderungen für die Produktionseinrichtung benötigen, siehe *Anpassen der Konfiguration*.
3. Erstellen Sie die Datenbank und ihre Struktur für Weblate (die Beispieleinstellungen verwenden PostgreSQL, sehen Sie sich *Datenbankeinrichtung für Weblate* für ein produktionsreifes Setup an):

```
weblate migrate
```

4. Erstellen Sie ein Benutzerkonto für den Administrator und kopieren Sie dessen Passwort in die Zwischenablage, und speichern Sie es zur späteren Verwendung:

```
weblate createadmin
```

5. Sammeln Sie die statischen Dateien für Ihren Webserver (siehe *Laufender Server* und *Bereitstellung statischer Dateien*):

```
weblate collectstatic
```

6. Komprimieren Sie die JavaScript- und CSS-Dateien (optional, siehe *Client-Assets komprimieren*):

```
weblate compress
```

7. Starten Sie die Celery-Arbeiter. Dies ist für Entwicklungszwecke nicht notwendig, wird aber ansonsten dringend empfohlen. *Hintergrundaufgaben mit Celery* hat mehr Informationen:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Starten Sie den Entwicklungsserver (:ref:'server' beschreibt ein Produktionssetup):

```
weblate runserver
```

Nach der Installation

Herzlichen Glückwunsch, Ihr Weblate-Server läuft jetzt und Sie können ihn nutzen.

- Sie können jetzt auf Weblate unter `http://localhost:8000/` zugreifen.
- Melden Sie sich mit den bei der Installation erhaltenen Administrator-Zugangsdaten an oder registrieren Sie sich mit neuen Benutzern.
- Sie können jetzt Weblate-Befehle mit dem Befehl **weblate** ausführen, wenn Weblate virtualenv aktiv ist, siehe *Management commands*.
- Sie können den Testserver mit Strg+C anhalten.
- Überprüfen Sie mögliche Probleme mit Ihrer Installation entweder unter der URL `/manage/performance/` (siehe *Verwaltungsoberfläche*) oder mit **weblate check --deploy**, siehe *Produktionseinrichtung*.

Hinzufügen einer Übersetzung

1. Öffnen Sie die Verwaltungsoberfläche (`http://localhost:8000/create/project/`) und erstellen Sie das Projekt, das Sie übersetzen möchten. Siehe *Projektkonfiguration* für weitere Details.

Sie müssen hier nur den Namen des Projekts und dessen Website angeben.

2. Erstellen Sie eine Komponente, die das eigentliche Objekt für die Übersetzung ist – sie verweist auf das VCS-Repository und wählt die zu übersetzenden Dateien aus. Siehe *Komponentenkonfiguration* für weitere Details.

Die wichtigen Felder hier, um übersetzbare Dateien zu finden, sind: *Name der Komponente*, *Quellcode-Repository* und *Dateimaske*. Weblate unterstützt eine Vielzahl von Formaten, darunter *GNU gettext*, *Android String-Ressourcen*, *Apple iOS-Zeichenketten*, *Java properties*, *Stringsdict-Format* oder *Fluent-Format*, siehe *Unterstützte Dateiformate* für weitere Details.

3. Sobald die Schritte oben abgeschlossen sind (je nach Größe Ihres VCS-Repositorys und der Anzahl der zu übersetzenden Nachrichten kann dies ein langwieriger Prozess sein), können Sie mit dem Übersetzen beginnen.

Installation unter macOS

Hardwareanforderungen

Weblate sollte auf jeder zeitgemäßen Hardware problemlos laufen. Nachfolgend finden Sie die minimale Konfiguration, die erforderlich ist, um Weblate auf einem einzelnen Host zu betreiben (Weblate, Datenbank und Webserver):

- 3 GB Arbeitsspeicher
- 2 CPU-Kerne
- 1 GB Speicherplatz

Je mehr Speicher, desto besser – er wird für das Caching auf allen Ebenen (Dateisystem, Datenbank und Weblate) verwendet.

Viele gleichzeitige Benutzer erhöhen die Anzahl der benötigten CPU-Kerne. Für Hunderte von Übersetzungskomponenten werden mindestens 4 GB RAM empfohlen.

Dies hat schwerwiegende Auswirkungen auf die Leistung der Weboberfläche und beeinträchtigt Funktionen, die von regelmäßigen Auslösern abhängen (z. B. das Übertragen ausstehender Änderungen, Digest-Benachrichtigungen oder Sicherungen).

Bemerkung: Die tatsächlichen Anforderungen an Ihre Weblate-Installation hängen stark von der Größe der darin verwalteten Übersetzungen ab.

Installation

Systemvoraussetzungen

Installieren Sie die Abhängigkeiten, die für die Erstellung der Python-Module erforderlich sind (siehe [Software-Anforderungen](#)):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip install virtualenv
```

Stellen Sie sicher, dass pip in der Lage ist, die libffi-Version zu finden, die von homebrew bereitgestellt wird – diese wird während des Installationsschritts benötigt.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig"
```

Installieren Sie die gewünschten optionalen Abhängigkeiten abhängig von den Funktionen, die Sie verwenden wollen (siehe [Optionale Abhängigkeiten](#)):

```
brew install tesseract
```

Installieren Sie optional Software für den Betrieb des Produktionsservers, siehe [Laufender Server](#), [Datenbankeinrichtung für Weblate](#), [Hintergrundaufgaben mit Celery](#). Je nach Größe Ihrer Installation sollten Sie diese Komponenten auf getrennten Servern laufen lassen.

Lokale Installationsanweisungen:

```
# Web server option 1: NGINX and uWSGI
brew install nginx uwsgi

# Web server option 2: Apache with ``mod_wsgi``
brew install httpd

# Caching backend: Redis
brew install redis

# Database server: PostgreSQL
brew install postgresql
```

Python-Module

Hinweis: Sie können Einstellungen vor oder nach der obigen Importanweisung definieren, um festzulegen, welche Einstellungen Vorrang haben. Vor der Importanweisung definierte Einstellungen können durch Umgebungsvariablen und im Daten-Volumen definierte Einstellungsüberschreibungen außer Kraft gesetzt werden. Einstellungen, die nach der Importanweisung definiert werden, können nicht überschrieben werden.

1. Erstellen Sie die Virtualenv für Weblate:

```
virtualenv ~/weblate-env
```

2. Aktivieren Sie die Virtualenv für Weblate:

```
. ~/weblate-env/bin/activate
```

3. Installieren Sie Weblate einschließlich aller optionalen Abhängigkeiten:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Bitte prüfen Sie *Optionale Abhängigkeiten* für die Feinabstimmung der optionalen Abhängigkeiten.

Bemerkung: Auf einigen Linux-Distributionen schlägt die Ausführung von Weblate mit einem libffi-Fehler fehl:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

Dies wird durch die Inkompatibilität von Binärpaketen verursacht, die über PyPI mit der Distribution verteilt werden. Um dies zu beheben, müssen Sie das Paket auf Ihrem System neu erstellen:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate konfigurieren

Bemerkung: Im Folgenden wird davon ausgegangen, dass die von Weblate verwendete Virtualenv aktiviert ist (durch Ausführung von `. ~/weblate-env/bin/activate`). Falls nicht, geben Sie den vollständigen Pfad zum Befehl **weblate** als `~/weblate-env/bin/weblate` an.

1. Kopieren Sie die Datei `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` nach `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Passen Sie die Werte in der neuen Datei `settings.py` nach Ihren Wünschen an. Sie müssen zumindest die Datenbank-Zugangsdaten und den geheimen Django-Schlüssel angeben, aber Sie werden mehr Änderungen für die Produktionseinrichtung benötigen, siehe *Anpassen der Konfiguration*.
3. Erstellen Sie die Datenbank und ihre Struktur für Weblate (die Beispieleinstellungen verwenden PostgreSQL, sehen Sie sich *Datenbankeinrichtung für Weblate* für ein produktionsreifes Setup an):

```
weblate migrate
```

4. Erstellen Sie ein Benutzerkonto für den Administrator und kopieren Sie dessen Passwort in die Zwischenablage, und speichern Sie es zur späteren Verwendung:

```
weblate createadmin
```

5. Sammeln Sie die statischen Dateien für Ihren Webserver (siehe *Laufender Server* und *Bereitstellung statischer Dateien*):

```
weblate collectstatic
```

6. Komprimieren Sie die JavaScript- und CSS-Dateien (optional, siehe *Client-Assets komprimieren*):

```
weblate compress
```

7. Starten Sie die Celery-Arbeiter. Dies ist für Entwicklungszwecke nicht notwendig, wird aber ansonsten dringend empfohlen. *Hintergrundaufgaben mit Celery* hat mehr Informationen:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Starten Sie den Entwicklungsserver (:ref:'server' beschreibt ein Produktionssetup):

```
weblate runserver
```

Nach der Installation

Herzlichen Glückwunsch, Ihr Weblate-Server läuft jetzt und Sie können ihn nutzen.

- Sie können jetzt auf Weblate unter `http://localhost:8000/` zugreifen.
- Melden Sie sich mit den bei der Installation erhaltenen Administrator-Zugangsdaten an oder registrieren Sie sich mit neuen Benutzern.
- Sie können jetzt Weblate-Befehle mit dem Befehl **weblate** ausführen, wenn Weblate virtualenv aktiv ist, siehe *Management commands*.
- Sie können den Testserver mit Strg+C anhalten.
- Überprüfen Sie mögliche Probleme mit Ihrer Installation entweder unter der URL `/manage/performance/` (siehe *Verwaltungsoberfläche*) oder mit **weblate check --deploy**, siehe *Produktionseinrichtung*.

Hinzufügen einer Übersetzung

1. Öffnen Sie die Verwaltungsoberfläche (`http://localhost:8000/create/project/`) und erstellen Sie das Projekt, das Sie übersetzen möchten. Siehe *Projektkonfiguration* für weitere Details.

Sie müssen hier nur den Namen des Projekts und dessen Website angeben.

2. Erstellen Sie eine Komponente, die das eigentliche Objekt für die Übersetzung ist – sie verweist auf das VCS-Repository und wählt die zu übersetzenden Dateien aus. Siehe *Komponentenkonfiguration* für weitere Details.

Die wichtigen Felder hier, um übersetzbare Dateien zu finden, sind: *Name der Komponente*, *Quellcode-Repository* und *Dateimaske*. Weblate unterstützt eine Vielzahl von Formaten, darunter *GNU gettext*, *Android String-Ressourcen*, *Apple iOS-Zeichenketten*, *Java properties*, *Stringsdict-Format* oder *Fluent-Format*, siehe *Unterstützte Dateiformate* für weitere Details.

3. Sobald die Schritte oben abgeschlossen sind (je nach Größe Ihres VCS-Repositorys und der Anzahl der zu übersetzenden Nachrichten kann dies ein langwieriger Prozess sein), können Sie mit dem Übersetzen beginnen.

Installieren aus Quellen

1. Bitte folgen Sie zunächst den Installationsanweisungen für Ihr System, bevor Sie Weblate installieren:
 - *Installation unter Debian und Ubuntu*
 - *Installation unter SUSE und openSUSE*
 - *Installation unter RedHat, Fedora und CentOS*
2. Schnappen Sie sich die neuesten Weblate-Quellen mittels Git (oder laden Sie einen Tarball herunter und packen Sie ihn aus):

```
git clone https://github.com/WeblateOrg/weblate.git weblate-src
```

Alternativ können Sie auch freigegebene Archive verwenden. Sie können sie von unserer Website <<https://weblate.org/>> herunterladen. Diese Downloads sind kryptografisch signiert, siehe *Überprüfen von Veröffentlichungssignaturen*.

3. Installieren Sie den aktuellen Weblate-Code in die virtualenv:

```
. ~/weblate-env/bin/activate
pip install -e weblate-src
```

4. Kopieren Sie `weblate/settings_example.py` nach `weblate/settings.py`.
5. Passen Sie die Werte in der neuen Datei `settings.py` nach Ihren Wünschen an. Sie müssen zumindest die Datenbank-Zugangsdaten und den geheimen Django-Schlüssel angeben, aber Sie werden mehr Änderungen für die Produktionseinrichtung benötigen, siehe *Anpassen der Konfiguration*.
6. Erstellen Sie die von Weblate verwendete Datenbank, siehe *Datenbankeinrichtung für Weblate*.
7. Erstellen Sie Django-Tabellen, statische Dateien und Anfangsdaten (siehe `:ref:'tables-setup'` und `:ref:'static-files'`):

```
weblate migrate
weblate collectstatic
weblate compress
```

Bemerkung: Dieser Schritt sollte jedes Mal wiederholt werden, wenn Sie das Repository aktualisieren.

Installation auf OpenShift

Mit dem Kubernetes Helm-Diagramm können Sie Ihre persönliche Weblate-Instanz in Sekundenschnelle zum Laufen bringen. Alle Abhängigkeiten von Weblate sind bereits enthalten. PostgreSQL ist als Standarddatenbank eingerichtet und es werden dauerhafte Volume-Claims verwendet.

Sie finden die Vorlage unter <https://github.com/WeblateOrg/openshift/>.

Installation

Die folgenden Beispiele gehen davon aus, dass Sie eine funktionierende OpenShift v3.x-Umgebung haben, in der das Client-Tool `oc` installiert ist. Bitte lesen Sie in der OpenShift-Dokumentation nach, um Anweisungen zu erhalten.

Die `template.yml` ist für den Betrieb aller Komponenten in OpenShift geeignet. Es gibt auch die Datei `template-external-postgresql.yml`, die keinen PostgreSQL-Server startet und es Ihnen ermöglicht, einen externen PostgreSQL-Server zu konfigurieren.

Webkonsole

Kopieren Sie die Rohdaten aus `template.yml` und importieren Sie sie in Ihr Projekt. Verwenden Sie dann die Schaltfläche `Create` in der OpenShift-Webkonsole, um Ihre Anwendung zu erstellen. Die Webkonsole wird Sie nach den Werten für alle von der Vorlage verwendeten Parameter fragen.

CLI

Um die Weblate-Vorlage in die Vorlagenbibliothek Ihres aktuellen Projekts hochzuladen, übergeben Sie die Datei `template.yml` mit dem folgenden Befehl:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
  -n <PROJECT>
```

Die Vorlage kann nun über die Webkonsole oder das CLI ausgewählt werden.

Parameter

Die Parameter, die Sie außer Kraft setzen können, sind im Abschnitt Parameter der Vorlage aufgeführt. Sie können sie mit der CLI auflisten, indem Sie den folgenden Befehl verwenden und die zu verwendende Datei angeben:

```
$ oc process --parameters -f https://raw.githubusercontent.com/WeblateOrg/
↪openshift/main/template.yml

# If the template is already uploaded
$ oc process --parameters -n <PROJECT> weblate
```

Bereitstellung

Sie können Vorlagen auch über die Befehlszeilenschnittstelle (CLI) verarbeiten und die erzeugte Konfiguration verwenden, um sofort Objekte zu erstellen.

```
$ oc process -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
  -p APPLICATION_NAME=weblate \
  -p WEBLATE_VERSION=4.3.1-1 \
  -p WEBLATE_SITE_DOMAIN=weblate.app-openshift.example.com \
  -p POSTGRESQL_IMAGE=docker-registry.default.svc:5000/openshift/postgresql:9.6 \
  -p REDIS_IMAGE=docker-registry.default.svc:5000/openshift/redis:3.2 \
  | oc create -f
```

Die Weblate-Instanz sollte nach erfolgreicher Migration und Bereitstellung unter dem angegebenen Parameter `WEBLATE_SITE_DOMAIN` verfügbar sein.

Nach der Container-Einrichtung können Sie sich als Benutzer *admin* mit dem in `WEBLATE_ADMIN_PASSWORD` bereitgestellten Passwort oder, falls es nicht festgelegt wurde, mit einem beim ersten Start erzeugten Zufallspasswort anmelden.

Um das *admin*-Passwort zurückzusetzen, starten Sie den Container neu, wobei `WEBLATE_ADMIN_PASSWORD` auf das neue Passwort im entsprechenden Secret gesetzt wird.

Beseitigen

```
$ oc delete all -l app=<APPLICATION_NAME>
$ oc delete configmap -l app= <APPLICATION_NAME>
$ oc delete secret -l app=<APPLICATION_NAME>
# ATTENTION! The following command is only optional and will permanently delete
↪all of your data.
$ oc delete pvc -l app=<APPLICATION_NAME>

$ oc delete all -l app=weblate \
  && oc delete secret -l app=weblate \
  && oc delete configmap -l app=weblate \
  && oc delete pvc -l app=weblate
```


Konfiguration

Durch die Verarbeitung der Vorlage wird eine entsprechende `ConfigMap` erstellt, die zur Anpassung des Weblate-Images verwendet werden kann. Die `ConfigMap` wird direkt als Umgebungsvariable eingebunden und löst bei jeder Änderung eine neue Bereitstellung aus. Für weitere Konfigurationsoptionen siehe [Docker-Umgebungsvariablen](#) für eine vollständige Liste der Umgebungsvariablen.

Unter Kubernetes installieren

Bemerkung: Diese Anleitung sucht Mitwirkende, die Erfahrung mit Kubernetes haben und die Einrichtung im Detail beschreiben.

Mit dem Kubernetes Helm-Diagramm können Sie Ihre persönliche Weblate-Instanz in Sekundenschnelle zum Laufen bringen. Alle Abhängigkeiten von Weblate sind bereits enthalten. PostgreSQL ist als Standarddatenbank eingerichtet und es werden dauerhafte Volume-Claims verwendet.

Sie finden das Diagramm unter <https://github.com/WeblateOrg/helm/> und es kann unter <https://artifacthub.io/packages/helm/weblate/weblate> angezeigt werden.

Installation

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Konfiguration

Für weitere Konfigurationsoptionen siehe [Docker-Umgebungsvariablen](#) für eine vollständige Liste der Umgebungsvariablen.

Wählen Sie je nach Ihren Gegebenheiten und Ihrer Erfahrung eine für Sie geeignete Installationsmethode:

- *Installation über Docker*, empfohlen für Produktionssysteme.
- Virtualenv-Installation, empfohlen für Produktionssysteme:
 - *Installation unter Debian und Ubuntu*
 - *Installation unter SUSE und openSUSE*
 - *Installation unter RedHat, Fedora und CentOS*
 - *Installation unter macOS*
- *Installieren aus Quellen*, empfohlen für die Entwicklung.
- *Installation auf OpenShift*
- *Unter Kubernetes installieren*

2.1.2 Software-Anforderungen

Betriebssystem

Von Weblate ist bekannt, dass es unter Linux, FreeBSD und macOS funktioniert. Andere Unix-ähnliche Systeme werden höchstwahrscheinlich auch funktionieren.

Weblate wird unter Windows nicht unterstützt. Aber es kann trotzdem funktionieren und Patches werden gerne angenommen.

Andere Dienste

Weblate nutzt andere Dienste für seinen Betrieb. Sie müssen mindestens die folgenden Dienste ausführen:

- PostgreSQL-Datenbank-Server, siehe *Datenbankeinrichtung für Weblate*.
- Redis-Server für Cache und Aufgabenwarteschlange, siehe *Hintergrundaufgaben mit Celery*.
- SMTP-Server für ausgehende E-Mails, siehe *Konfigurieren ausgehender E-Mail*.

Python-Abhängigkeiten

Weblate ist in *Python* geschrieben und unterstützt Python 3.6 oder neuer. Sie können die Abhängigkeiten mit pip oder aus Ihren Distributionspaketen installieren, eine vollständige Liste finden Sie in `requirements.txt`.

Wichtigste Abhängigkeiten:

Django

<https://www.djangoproject.com/>

Celery

<https://docs.celeryq.dev/>

Translate Toolkit

<https://toolkit.translatehouse.org/>

translation-finder

<https://github.com/WeblateOrg/translation-finder>

Python Social Auth

<https://python-social-auth.readthedocs.io/>

Django REST-Framework

<https://www.django-rest-framework.org/>

Optionale Abhängigkeiten

Die folgenden Module sind für einige Weblate-Funktionen erforderlich. Sie finden sie alle in `requirements-optional.txt`.

Mercurial (optional für die Unterstützung von `:ref:`vcs-mercurial-Repositorys`)`

<https://www.mercurial-scm.org/>

phply (optional für *PHP-Zeichenketten*)

<https://github.com/viraptor/phply>

tesseract (optional für OCR in Bildschirmfotos)

<https://github.com/sirfz/tesseract>

python-akismet (optional für *Spam protection*)

<https://github.com/Nekmo/python-akismet>

ruamel.yaml (optional für *YAML-Dateien*)

<https://pypi.org/project/ruamel.yaml/>

zeep (optional für *Microsoft Terminology*)

<https://docs.python-zeep.org/>

aeidon (optional für *Untertiteldateien*)

<https://pypi.org/project/aeidon/>

fluent.syntax (optional für *Fluent-Format*)

<https://projectfluent.org/>

Hinweis: Wenn Sie mit pip installieren, können Sie die gewünschten Funktionen direkt bei der Installation angeben:

```
pip install "Weblate[PHP,Fluent]"
```

Oder Sie können Weblate mit allen optionalen Funktionen installieren:

```
pip install "Weblate[all]"
```

Oder Sie können Weblate ohne optionale Funktionen installieren:

```
pip install Weblate
```

Datenbank-Backend-Abhängigkeiten

Weblate unterstützt PostgreSQL, MySQL und MariaDB, siehe *Datenbankeinrichtung für Weblate* und Backend-Dokumentation für weitere Details.

Weitere Systemanforderungen

Die folgenden Abhängigkeiten müssen auf dem System installiert sein:

Git

<https://git-scm.com/>

Pango, Cairo und zugehörige Header-Dateien sowie GObject-Introspektionsdaten

<https://cairographics.org/>, <https://pango.gnome.org/>, siehe Pangokairo

git-review (optional für Gerrit-Unterstützung)

<https://pypi.org/project/git-review/>

git-svn (optional für Subversion-Unterstützung)

<https://git-scm.com/docs/git-svn>

tesseract und seine Daten (optional für Bildschirmfotos-OCR)

<https://github.com/tesseract-ocr/tesseract>

licensee (optional zur Erkennung der Lizenz beim Erstellen der Komponente)

<https://github.com/licensee/licensee>

Build-Zeit-Abhängigkeiten

Um einige der *Python-Abhängigkeiten* zu bauen, müssen Sie möglicherweise deren Abhängigkeiten installieren. Dies hängt davon ab, wie Sie sie installieren, also konsultieren Sie bitte die Dokumentation der einzelnen Pakete. Sie werden diese nicht benötigen, wenn Sie das vorgefertigte `Wheels` verwenden, während Sie `pip` installieren oder wenn Sie Distributionspakete verwenden.

Pango und Cairo

Geändert in Version 3.7.

Weblate verwendet Pango und Cairo für die Darstellung von Bitmap-Widgets (siehe *promotion*) und für die Darstellung von Qualitätsprüfungen (siehe *Schriftarten verwalten*). Um die Python-Bindings dafür richtig zu installieren, müssen Sie zuerst die Systembibliotheken installieren – Sie brauchen sowohl Cairo als auch Pango, die wiederum GLib benötigen. Alle diese Bibliotheken sollten zusammen mit Entwicklungsdateien und GObject-Introspektionsdaten installiert werden.

2.1.3 Überprüfen von Veröffentlichungssignaturen

Weblate-Veröffentlichungen werden vom veröffentlichenden Entwickler kryptografisch signiert. Derzeit ist dies Michal Čihař. Der Fingerabdruck seines PGP-Schlüssels lautet:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

und weitere Informationen zur Identifizierung erhalten Sie unter [<https://keybase.io/nijel>](https://keybase.io/nijel).

Sie sollten überprüfen, ob die Signatur mit dem heruntergeladenen Archiv übereinstimmt. Auf diese Weise können Sie sicher sein, dass Sie denselben Code verwenden, der veröffentlicht wurde. Sie sollten auch das Datum der Signatur überprüfen, um sicherzustellen, dass Sie die neueste Version heruntergeladen haben.

Jedem Archiv liegen `.asc` Dateien bei, die die PGP-Signatur für das Archiv enthalten. Sobald Sie beide Dateien im selben Ordner haben, können Sie die Signatur überprüfen:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

Wie Sie sehen können, beschwert sich GPG, dass es den öffentlichen Schlüssel nicht kennt. An diesem Punkt sollten Sie einen der folgenden Schritte durchführen:

- Verwenden Sie `wkd`, um den Schlüssel herunterzuladen:

```
$ gpg --auto-key-locate wkd --locate-keys michal@cihar.com
pub   rsa4096 2009-06-17 [SC]
      63CB1DF1EF12CF2AC0EE5A329C27B31342B7511D
uid           [ultimate] Michal Čihař <michal@cihar.com>
uid           [ultimate] Michal Čihař <nijel@debian.org>
uid           [ultimate] [jpeg image of size 8848]
uid           [ultimate] Michal Čihař (Braiiins) <michal.cihar@braiins.cz>
sub   rsa4096 2009-06-17 [E]
sub   rsa4096 2015-09-09 [S]
```

- Laden Sie den Keyring von [Michals Server](#) herunter und importieren Sie ihn mit:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Herunterladen und Importieren des Schlüssels von einem der Schlüsselservers:

```
$ gpg --keyserver hkp://pgp.mit.edu --recv-keys 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: key 9C27B31342B7511D: "Michal Čihař <michal@cihar.com>" imported
gpg: Total number processed: 1
gpg:                unchanged: 1
```

Dies verbessert die Situation ein wenig – zu diesem Zeitpunkt können Sie zwar überprüfen, ob die Signatur des angegebenen Schlüssels korrekt ist, aber Sie können dem im Schlüssel verwendeten Namen immer noch nicht vertrauen:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:                aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:                aka "[jpeg image of size 8848]" [ultimate]
gpg:                aka "Michal Čihař (Brains) <michal.cihar@brains.cz>" [ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE  5A32 9C27 B313 42B7 511D
```

Das Problem dabei ist, dass jeder einen Schlüssel mit diesem Namen ausstellen könnte. Sie müssen sicherstellen, dass der Schlüssel tatsächlich der genannten Person gehört. Das GNU Privacy Handbook behandelt dieses Thema im Kapitel [Validating other keys on your public keyring](#). Die zuverlässigste Methode ist es, den Entwickler persönlich zu treffen und die Fingerabdrücke der Schlüssel auszutauschen, aber Sie können sich auch auf das Netz des Vertrauens verlassen. Auf diese Weise können Sie dem Schlüssel durch die Unterschriften anderer Personen, die den Entwickler persönlich getroffen haben, vertrauen.

Sobald der Schlüssel vertrauenswürdig ist, wird die Warnung nicht angezeigt:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:                aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:                aka "[jpeg image of size 8848]" [ultimate]
gpg:                aka "Michal Čihař (Brains) <michal.cihar@brains.cz>" [ultimate]
```

Sollte die Signatur ungültig sein (das Archiv wurde geändert), erhalten Sie eine eindeutige Fehlermeldung, unabhängig davon, ob der Schlüssel vertrauenswürdig ist oder nicht:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: BAD signature from "Michal Čihař <michal@cihar.com>" [ultimate]
```

2.1.4 Dateisystemberechtigungen

Der Weblate-Prozess muss in der Lage sein, vom und ins Verzeichnis zu lesen und zu schreiben, in dem er Daten speichert – `DATA_DIR`. Alle Dateien in diesem Verzeichnis sollten dem Benutzer gehören, der alle Weblate-Prozesse ausführt (normalerweise WSGI und Celery, siehe [Laufender Server](#) und [Hintergrundaufgaben mit Celery](#)).

Die Standardkonfiguration platziert sie im selben Baum wie die Weblate-Quellen, aber Sie könnten es vorziehen, sie an einen besseren Ort zu verschieben, wie zum Beispiel: `/var/lib/weblate`.

Weblate versucht, diese Verzeichnisse automatisch zu erstellen, aber es wird scheitern, wenn es keine entsprechenden Rechte hat.

Sie sollten auch vorsichtig sein, wenn Sie *Management commands* ausführen, da sie unter demselben Benutzer ausgeführt werden sollten, unter dem auch Weblate selbst läuft, da sonst die Berechtigungen für einige Dateien falsch sein könnten.

Im Docker-Container müssen alle Dateien im Volume `/app/data` dem Benutzer `weblate` innerhalb des Containers gehören (UID 1000).

Siehe auch:

Bereitstellung statischer Dateien

2.1.5 Datenbankeinrichtung für Weblate

Es wird empfohlen, Weblate mit einem PostgreSQL-Datenbankserver zu betreiben.

Siehe auch:

Eine leistungsstarke Datenbank-Engine verwenden, Databases, Migration von anderen Datenbanken zu PostgreSQL

PostgreSQL

PostgreSQL ist normalerweise die beste Wahl für Django-basierte Websites. Es ist die Referenzdatenbank, die für die Implementierung der Django-Datenbankschicht verwendet wird.

Bemerkung: Weblate verwendet die Trigram-Erweiterung, die in einigen Fällen separat installiert werden muss. Suchen Sie nach `postgresql-contrib` oder einem ähnlich benannten Paket.

Siehe auch:

[PostgreSQL notes](#)

Erstellen einer Datenbank in PostgreSQL

Normalerweise ist es eine gute Idee, Weblate in einer separaten Datenbank und unter einem separaten Benutzerkonto laufen zu lassen:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Hinweis: Wenn Sie den Weblate-Benutzer nicht zu einem Superuser in PostgreSQL machen wollen, können Sie das weglassen. In diesem Fall müssen Sie einige der Migrationsschritte manuell als PostgreSQL-Superuser im Schema durchführen, das Weblate verwenden wird:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
CREATE EXTENSION IF NOT EXISTS btree_gin WITH SCHEMA weblate;
```

Weblate für die Verwendung von PostgreSQL konfigurieren

Das `settings.py`-Snippet für PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

Die Datenbankmigration führt `ALTER ROLE` für die von Weblate verwendete Datenbankrolle durch. In den meisten Fällen stimmt der Name der Rolle mit dem Benutzernamen überein. In komplexeren Setups unterscheidet sich der Rollename vom Benutzernamen und Sie erhalten eine Fehlermeldung über eine nicht existierende Rolle während der Datenbankmigration (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Es ist bekannt, dass dies mit Azure Database for PostgreSQL passiert, aber es ist nicht auf diese Umgebung beschränkt. Bitte setzen Sie `ALTER_ROLE`, um den Namen der Rolle zu ändern, die Weblate während der Datenbankmigration ändern soll.

MySQL und MariaDB

Hinweis: Einige Weblate-Funktionen werden mit *PostgreSQL* besser funktionieren. Dazu gehören die Suche und das Translation Memory, die Volltextfunktionen in der Datenbank nutzen, wobei die PostgreSQL-Implementierung besser ist.

Weblate kann auch mit MySQL oder MariaDB verwendet werden, bitte lesen Sie [MySQL notes](#) und [MariaDB notes](#) für Hinweise zur Verwendung von Django mit diesen. Aufgrund der Einschränkungen wird empfohlen, *PostgreSQL* für neue Installationen zu verwenden.

Weblate benötigt MySQL mindestens in der Version 5.7.8 oder MariaDB mindestens in der Version 10.2.7.

Die folgende Konfiguration wird für Weblate empfohlen:

- Verwenden Sie den Zeichensatz `utf8mb4`, um die Darstellung höherer Unicode-Ebenen (zum Beispiel Emojis) zu ermöglichen.
- Konfigurieren Sie den Server mit `innodb_large_prefix`, um längere Indizes für Textfelder zu ermöglichen.
- Setzen Sie die Isolationsstufe auf `READ COMMITTED`.
- Der SQL-Modus sollte auf `STRICT_TRANS_TABLES` eingestellt werden.

MySQL 8.x, MariaDB 10.5.x oder neuer haben eine vernünftige Standardkonfiguration, sodass keine Serveranpassung notwendig sein sollte und alles, was benötigt wird, auf der Client-Seite konfiguriert werden kann.

Unten ist ein `/etc/my.cnf.d/server.cnf` Beispiel für einen Server mit 8 GB RAM. Diese Einstellungen sollten für die meisten Installationen ausreichend sein. MySQL und MariaDB verfügen über Einstellungen, die die

Leistung Ihres Servers erhöhen, die aber nicht als notwendig erachtet werden, es sei denn, Sie planen, dass eine große Anzahl von Benutzern gleichzeitig auf das System zugreift. Einzelheiten hierzu finden Sie in der Dokumentation des jeweiligen Anbieters.

Um Probleme bei der Installation zu vermeiden, ist es absolut wichtig, dass die Einstellung `innodb_file_per_table` richtig gesetzt ist und MySQL/MariaDB neu gestartet wurde, bevor Sie Ihre Weblate-Installation starten.

```
[mysqld]
character-set-server = utf8mb4
character-set-client = utf8mb4
collation-server = utf8mb4_unicode_ci

datadir=/var/lib/mysql

log-error=/var/log/mariadb/mariadb.log

innodb_large_prefix=1
innodb_file_format=Barracuda
innodb_file_per_table=1
innodb_buffer_pool_size=2G
sql_mode=STRICT_TRANS_TABLES
```

Hinweis: Falls Sie den Fehler #1071 – Specified key was too long; max key length is 767 bytes erhalten, aktualisieren Sie bitte Ihre Konfiguration, um die obigen Einstellungen `innodb` zu übernehmen und starten Sie Ihre Installation neu.

Hinweis: Falls Sie den Fehler #2006 – MySQL server has gone away erhalten, könnte die Konfiguration von `CONN_MAX_AGE` helfen.

Weblate für die Verwendung von MySQL/MariaDB konfigurieren

Das `settings.py`-Snippet für MySQL und MariaDB:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "127.0.0.1",
        # Set to empty string for default
        "PORT": "3306",
        # In case you wish to use additional
        # connection options
        "OPTIONS": {},
    }
}
```

Sie sollten auch das Benutzerkonto `weblate` in MySQL oder MariaDB erstellen, bevor Sie mit der Installation beginnen. Verwenden Sie dazu die folgenden Befehle:


```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Andere Konfigurationen

Konfigurieren ausgehender E-Mail

Weblate verschickt zu verschiedenen Anlässen E-Mails – zur Kontoaktivierung und zu verschiedenen vom Nutzer konfigurierten Benachrichtigungen. Hierfür benötigt es Zugang zu einem SMTP-Server.

Das Mailserver-Setup wird mit diesen Einstellungen konfiguriert: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_TLS`, `EMAIL_USE_SSL`, `EMAIL_HOST_USER` und `EMAIL_PORT`. Ihre Namen sind ziemlich selbsterklärend, aber Sie können mehr Informationen in der Django-Dokumentation finden.

Hinweis: Falls Sie die Fehlermeldung erhalten, dass die Authentifizierung nicht unterstützt wird (z. B. `SMTP AUTH extension not supported by server`), liegt das wahrscheinlich daran, dass Sie eine unsichere Verbindung verwenden und der Server sich weigert, sich auf diese Weise zu authentifizieren. Versuchen Sie in diesem Fall `EMAIL_USE_TLS` zu aktivieren.

Siehe auch:

Keine E-Mails von Weblate erhalten, Konfiguration der ausgehenden E-Mails im Docker-Container

Hinter einem Reverse-Proxy ausführen

Mehrere Funktionen in Weblate sind darauf angewiesen, dass die IP-Adresse des Clients ermittelt werden kann. Dazu gehören *Ratenbegrenzung*, *Spam protection* oder *Audit-Protokoll*.

In der Standardkonfiguration parst Weblate die IP-Adresse von `REMOTE_ADDR`, die durch den WSGI-Handler gesetzt wird.

Falls Sie einen Reverse-Proxy verwenden, enthält dieses Feld höchstwahrscheinlich dessen Adresse. Sie müssen Weblate so konfigurieren, dass es zusätzlichen HTTP-Headern vertraut und die IP-Adresse aus diesen auswertet. Dies kann nicht standardmäßig aktiviert werden, da dies bei Installationen, die keinen Reverse-Proxy verwenden, IP-Adress-Spoofing ermöglichen würde. Die Aktivierung von `IP_BEHIND_REVERSE_PROXY` dürfte für die meisten üblichen Installationen ausreichen, aber möglicherweise müssen Sie auch `IP_PROXY_HEADER` und `IP_PROXY_OFFSET` anpassen.

Eine weitere Sache, auf die Sie achten müssen, ist der `Host`-Header. Er sollte mit dem übereinstimmen, der als `SITE_DOMAIN` konfiguriert ist. Möglicherweise ist eine zusätzliche Konfiguration in Ihrem Reverse-Proxy erforderlich (verwenden Sie zum Beispiel `ProxyPreserveHost On` für Apache oder `proxy_set_header Host $host;` für nginx).

Siehe auch:

Spam protection, *Ratenbegrenzung*, *Audit-Protokoll*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

HTTP-Proxy

Weblate führt VCS-Befehle aus und diese akzeptieren die Proxy-Konfiguration aus der Umgebung. Es wird empfohlen, die Proxy-Einstellungen in `settings.py` zu definieren:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Siehe auch:

[Proxy-Umgebungsvariablen](#)

2.1.7 Anpassen der Konfiguration

Siehe auch:

Sample configuration

Kopieren Sie `weblate/settings_example.py` nach `weblate/settings.py` und passen Sie es an Ihre Einrichtung an. Wahrscheinlich werden Sie die folgenden Optionen anpassen: `ADMINS`

Liste der Website-Administratoren, die Benachrichtigungen erhalten sollen, wenn etwas schief läuft, z. B. Benachrichtigungen über fehlgeschlagene Merges oder Django-Fehler.

Siehe auch:

`ADMINS`, *Administratoren richtig konfigurieren*

`ALLOWED_HOSTS`

Sie müssen hier die Hosts auflisten, die Ihre Website bedienen soll. Zum Beispiel:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternativ können Sie auch einen Platzhalter einfügen:

```
ALLOWED_HOSTS = ["*"]
```

Siehe auch:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Einrichtung zulässiger Hosts*

`SESSION_ENGINE`

Konfigurieren Sie, wie Ihre Sitzungen gespeichert werden sollen. Wenn Sie die Standard-Datenbank-Backend-Engine beibehalten, sollten Sie Folgendes vorsehen: **`weblate clearsessions`**, um veraltete Sitzungsdaten aus der Datenbank zu entfernen.

Wenn Sie Redis als Cache verwenden (siehe *Caching einschalten*), ist es empfehlenswert, ihn auch für Sitzungen zu verwenden:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Siehe auch:

Configuring the session engine, `SESSION_ENGINE`

`DATABASES`

Verbindung zum Datenbankserver, bitte lesen Sie die Django-Dokumentation für weitere Details.

Siehe auch:

Datenbankeinrichtung für Weblate, `DATABASES`, *Databases*

DEBUG

Deaktivieren Sie dies für alle Produktionsserver. Wenn der Debugmodus aktiviert ist, zeigt Django den Benutzern im Falle eines Fehlers Backtraces an. Wenn Sie ihn deaktivieren, werden Fehler per E-Mail an ADMINS (siehe oben) gesendet.

Der Debugmodus verlangsamt auch Weblate, da Django in diesem Fall viel mehr Informationen intern speichert.

Siehe auch:

DEBUG, *Debugmodus deaktivieren*

DEFAULT_FROM_EMAIL

E-Mail-Absenderadresse für ausgehende E-Mails, z. B. Registrierungs-E-Mails.

Siehe auch:

DEFAULT_FROM_EMAIL

SECRET_KEY

Schlüssel, der von Django verwendet wird, um einige Informationen in Cookies zu signieren, siehe *Geheimer Django-Schlüssel* für weitere Informationen.

Siehe auch:

SECRET_KEY

SERVER_EMAIL

E-Mail, die als Absenderadresse für den Versand von E-Mails an den Administrator verwendet wird, z. B. für Benachrichtigungen über fehlgeschlagene Merges.

Siehe auch:

SERVER_EMAIL

2.1.8 Füllen der Datenbank

Nachdem Ihre Konfiguration fertig ist, können Sie `weblate migrate` ausführen, um die Datenbankstruktur zu erstellen. Nun sollten Sie in der Lage sein, Übersetzungsprojekte über die Adminoberfläche zu erstellen.

Wenn Sie die Installation nicht interaktiv durchführen möchten, können Sie `weblate migrate --noinput` verwenden und dann mit dem Befehl `createadmin` einen Administrator-Benutzer erstellen.

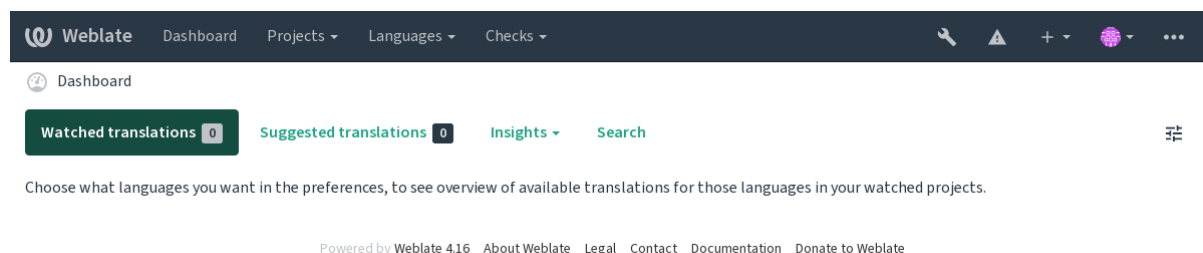
Wenn Sie fertig sind, sollten Sie auch den *Leistungsbericht* in der Adminoberfläche überprüfen, der Ihnen Hinweise auf eine möglicherweise nicht optimale Konfiguration Ihrer Website gibt.

Siehe auch:

Konfiguration, Liste der Berechtigungen und integrierten Rollen

2.1.9 Produktionseinrichtung

Für eine Produktionseinrichtung sollten Sie die in den folgenden Abschnitten beschriebenen Anpassungen vornehmen. Die kritischsten Einstellungen lösen eine Warnung aus, die durch ein Ausrufezeichen in der oberen Leiste angezeigt wird, wenn Sie als Superuser angemeldet sind:



Es wird auch empfohlen, die von Django ausgelösten Prüfungen zu kontrollieren (auch wenn Sie nicht alle korrigieren müssen):

```
weblate check --deploy
```

Sie können die gleiche Checkliste auch über die [Verwaltungsoberfläche](#) einsehen.

Siehe auch:

[Deployment checklist](#)

Debugmodus deaktivieren

Deaktivieren Sie den Debugmodus von Django (*DEBUG*) durch:

```
DEBUG = False
```

Wenn der Debugmodus aktiviert ist, speichert Django alle ausgeführten Abfragen und zeigt den Benutzern Rückverfolgungen von Fehlern an, was in einer Produktionsumgebung nicht erwünscht ist.

Siehe auch:

[Anpassen der Konfiguration](#)

Administratoren richtig konfigurieren

Setzen Sie die richtigen Administrator-Adressen in der Einstellung *ADMINS*, um festzulegen, wer E-Mails erhalten soll, wenn zum Beispiel auf dem Server etwas schief läuft:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Siehe auch:

[Anpassen der Konfiguration](#)

Seitendomain richtig einstellen

Passen Sie den Seitenname und die Seitendomain in der Adminoberfläche an, da sonst die Links in RSS oder Registrierungs-E-Mails nicht funktionieren. Dies wird mit *SITE_DOMAIN* konfiguriert, das den Namen der Seitendomain enthalten sollte.

Geändert in Version 4.2: Vor der Version 4.2 wurde stattdessen das Django Sites Framework verwendet, siehe [The "sites" framework](#).

Siehe auch:

[Einrichtung zulässiger Hosts](#), [HTTPS richtig konfigurieren](#) *SITE_DOMAIN*, *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

HTTPS richtig konfigurieren

Es wird dringend empfohlen, Weblate über das verschlüsselte HTTPS-Protokoll laufen zu lassen. Nachdem Sie es aktiviert haben, sollten Sie `ENABLE_HTTPS` in den Einstellungen setzen:

```
ENABLE_HTTPS = True
```

Hinweis: Vielleicht möchten Sie auch HSTS einrichten, siehe [SSL/HTTPS](#) für weitere Details.

Siehe auch:

[ENABLE_HTTPS](#), [Einrichtung zulässiger Hosts](#), [Seitendomain richtig einstellen](#)

SECURE_HSTS_SECONDS richtig einstellen

Wenn Ihre Website über SSL bereitgestellt wird, müssen Sie einen Wert für `SECURE_HSTS_SECONDS` in der `settings.py` setzen, um HTTP Strict Transport Security zu aktivieren. Standardmäßig ist dieser Wert auf 0 gesetzt, wie unten gezeigt.

```
SECURE_HSTS_SECONDS = 0
```

Wenn auf einen Integer-Wert ungleich Null gesetzt, setzt `django.middleware.security.SecurityMiddleware` den HTTP Strict Transport Security-Header auf alle Antworten, die ihn nicht bereits haben.

Warnung: Eine falsche Einstellung kann Ihre Website unwiderruflich (für einige Zeit) zerstören. Lesen Sie zuerst die [HTTP Strict Transport Security Dokumentation](#).

Eine leistungsstarke Datenbank-Engine verwenden

- Bitte verwenden Sie PostgreSQL für eine Produktionsumgebung, siehe [Datenbankeinrichtung für Weblate](#) für weitere Informationen.
- Verwenden Sie einen benachbarten Standort für den Betrieb des Datenbankservers, da sonst die Netzwerkleistung oder -zuverlässigkeit Ihr Weblate-Erlebnis beeinträchtigen könnte.
- Überprüfen Sie die Leistung des Datenbankservers oder passen Sie seine Konfiguration an, zum Beispiel mit PG Tune.

Siehe auch:

[Datenbankeinrichtung für Weblate](#), [Migration von anderen Datenbanken zu PostgreSQL](#), [Anpassen der Konfiguration, Databases](#)

Caching einschalten

Wenn möglich, verwenden Sie Redis von Django aus, indem Sie zum Beispiel die Konfigurationsvariable `CACHES` anpassen:

```
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/0",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # 'LOCATION': 'unix:///var/run/redis/redis.sock?db=0',
    },
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
"OPTIONS": {
    "CLIENT_CLASS": "django_redis.client.DefaultClient",
    "PARSER_CLASS": "redis.connection.HiredisParser",
},
}
```

Hinweis: Falls Sie die Redis-Einstellungen für den Cache ändern, müssen Sie diese möglicherweise auch für Celery anpassen, siehe [Hintergrundaufgaben mit Celery](#).

Siehe auch:

Avatar-Caching, Django's cache framework

Avatar-Caching

Zusätzlich zum Caching von Django führt Weblate auch ein Caching von Avataren durch. Es wird empfohlen, für diesen Zweck einen separaten, dateibasierten Cache zu verwenden:

```
CACHES = {
    "default": {
        # Default caching backend setup, see above
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "unix:///var/run/redis/redis.sock?db=0",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 604800,
        "OPTIONS": {
            "MAX_ENTRIES": 1000,
        },
    },
}
```

Siehe auch:

ENABLE_AVATARS, *AVATAR_URL_PREFIX*, *Avatars*, *Caching einschalten*, Django's cache framework

E-Mail-Versand konfigurieren

Weblate verschickt zu verschiedenen Anlässen E-Mails und diese sollten eine korrekte Absenderadresse haben. Bitte konfigurieren Sie *SERVER_EMAIL* und *DEFAULT_FROM_EMAIL* so, dass sie zu Ihrer Einsatzumgebung passen, zum Beispiel:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Bemerkung: Um den Versand von E-Mails durch Weblate zu deaktivieren, setzen Sie *EMAIL_BACKEND* auf `django.core.mail.backends.dummy.EmailBackend`.

Dies deaktiviert *alle* E-Mails, einschließlich der E-Mails zur Registrierung oder zum Zurücksetzen des Passworts.

Siehe auch:

Anpassen der Konfiguration, Konfigurieren ausgehender E-Mail, EMAIL_BACKEND, DEFAULT_FROM_EMAIL, SERVER_EMAIL

Einrichtung zulässiger Hosts

Django benötigt `ALLOWED_HOSTS`, um eine Liste von Domännennamen zu speichern, die Ihre Seite bedienen darf. Wenn Sie die Liste leer lassen, werden alle Anfragen blockiert.

Wenn dies nicht so konfiguriert ist, dass es zu Ihrem HTTP-Server passt, erhalten Sie Fehler wie Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.

Hinweis: Auf Docker-Containern ist dies als `WEBLATE_ALLOWED_HOSTS` verfügbar.

Siehe auch:

ALLOWED_HOSTS, WEBLATE_ALLOWED_HOSTS, Seitendomain richtig einstellen

Geheimer Django-Schlüssel

Die Einstellung `SECRET_KEY` wird von Django verwendet, um Cookies zu signieren. Sie sollten wirklich Ihren eigenen Wert erzeugen, anstatt den aus dem Beispiel-Setup zu verwenden.

Sie können einen neuen Schlüssel mittels `weblate-generate-secret-key` erzeugen, das mit Weblate ausgeliefert wird.

Siehe auch:

SECRET_KEY

Ausführen von Wartungsaufgaben

Für eine optimale Leistung empfiehlt es sich, einige Wartungsaufgaben im Hintergrund laufen zu lassen. Dies wird nun automatisch von *Hintergrundaufgaben mit Celery* erledigt und umfasst die folgenden Aufgaben:

- Konfigurationsintegritätsprüfung (stündlich).
- Übertragen anstehender Änderungen (stündlich), siehe *Lazy Commits* und `commit_pending`.
- Aktualisieren von Komponentenwarnungen (täglich).
- Entfernte Branches aktualisieren (nachts), siehe `AUTO_UPDATE`.
- Translation-Memory-Backup in JSON (täglich), siehe `dump_memory`.
- Volltext- und Datenbankwartungsaufgaben (tägliche und wöchentliche Aufgaben), siehe `:d jad-min:cleanuptrans`.

Geändert in Version 3.2: Seit Version 3.2 werden diese Aufgaben standardmäßig mit Celery ausgeführt und Weblate wird bereits mit einer entsprechenden Konfiguration geliefert, siehe *Hintergrundaufgaben mit Celery*.

System-Sprachumgebungen und -Kodierung

Die System-Sprachumgebungen sollten als UTF-8-fähig konfiguriert werden. Bei den meisten Linux-Distributionen ist dies die Standardeinstellung. Falls dies auf Ihrem System nicht der Fall ist, ändern Sie bitte die Sprachumgebungen auf die UTF-8-Variante.

Zum Beispiel, indem Sie `/etc/default/locale` bearbeiten und dort `LANG="C.UTF-8"` einstellen.

In einigen Fällen haben die einzelnen Dienste eine separate Konfiguration für Sprachumgebungen. Dies ist je nach Distribution und Webserver unterschiedlich, daher sollten Sie die Dokumentation Ihrer Webserver-Pakete daraufhin überprüfen.

Apache unter Ubuntu verwendet `/etc/apache2/envvars`:

```
export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'
```

Apache unter CentOS verwendet `/etc/sysconfig/httpd` (oder `/opt/rh/httpd24/root/etc/sysconfig/httpd`):

```
LANG='en_US.UTF-8'
```

Verwenden der benutzerdefinierten Zertifizierungsstelle

Weblate prüft SSL-Zertifikate bei HTTP-Anfragen. Wenn Sie eine benutzerdefinierte Zertifizierungsstelle verwenden, der in den Standardpaketen nicht vertraut wird, müssen Sie ihr Zertifikat als vertrauenswürdig hinzufügen.

Der bevorzugte Ansatz ist, dies auf Systemebene zu tun. Bitte lesen Sie die Dokumentation Ihrer Distribution für weitere Details (unter Debian kann dies zum Beispiel durch das Ablegen des CA-Zertifikats in `/usr/local/share/ca-certificates/` und dem Ausführen von `update-ca-certificates` geschehen).

Sobald dies geschehen ist, vertrauen die Systemprogramme dem Zertifikat, so auch Git.

Für Python-Code müssen Sie die Anfragen so konfigurieren, dass das System-CA-Bundle anstelle des mitgelieferten Pakets verwendet wird. Dies kann erreicht werden, indem der folgende Ausschnitt in `settings.py` eingefügt wird (der Pfad ist Debian-spezifisch):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Client-Assets komprimieren

Weblate wird mit einer Reihe von JavaScript- und CSS-Dateien geliefert. Aus Leistungsgründen ist es sinnvoll, diese vor dem Senden an einen Client zu komprimieren. In der Standardkonfiguration wird dies im laufenden Betrieb auf Kosten eines geringen Overheads durchgeführt. Bei großen Installationen empfiehlt es sich, den Offline-Komprimierungsmodus zu aktivieren. Dies muss in der Konfiguration vorgenommen werden, und die Komprimierung muss bei jeder Aktualisierung von Weblate ausgelöst werden.

Der Konfigurationswechsel ist einfach, indem man `django.conf.settings.COMPRESS_OFFLINE` aktiviert und `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` konfiguriert (letzteres ist bereits in der Beispielkonfiguration enthalten):

```
COMPRESS_OFFLINE = True
```

Bei jeder Bereitstellung müssen Sie die Dateien komprimieren, damit sie der aktuellen Version entsprechen:

```
weblate compress
```

Hinweis: Im offiziellen Docker-Image ist diese Funktion bereits aktiviert.

Siehe auch:

Common Deployment Scenarios, *Bereitstellung statischer Dateien*

2.1.10 Laufender Server

Hinweis: Falls Sie keine Erfahrung mit den unten beschriebenen Diensten haben, sollten Sie *Installation über Docker* ausprobieren.

Sie benötigen mehrere Dienste, um Weblate auszuführen, die empfohlene Einrichtung besteht aus:

- Datenbankserver (siehe *Datenbankeinrichtung für Weblate*)
- Cache-Server (siehe *Caching einschalten*)
- Frontend-Webserver für statische Dateien und SSL-Terminierung (siehe *Bereitstellung statischer Dateien*)
- WSGI-Server für dynamische Inhalte (siehe *Beispielkonfiguration für NGINX und uWSGI*)
- Celery für die Ausführung von Hintergrundaufgaben (siehe *Hintergrundaufgaben mit Celery*)

Bemerkung: Es gibt einige Abhängigkeiten zwischen den Diensten, zum Beispiel sollten Cache und Datenbank beim Starten von Celery- oder uwsgi-Prozessen bereits laufen.

In den meisten Fällen werden Sie alle Dienste auf einem einzigen (virtuellen) Server laufen lassen, aber wenn Ihre Installation stark ausgelastet ist, können Sie die Dienste aufteilen. Die einzige Einschränkung dabei ist, dass Celery- und Wsgi-Server Zugriff auf `DATA_DIR` benötigen.

Bemerkung: Der WSGI-Prozess muss unter demselben Benutzer wie der Celery-Prozess ausgeführt werden, sonst werden die Dateien im `DATA_DIR` mit unterschiedlichen Besitzverhältnissen gespeichert, was zu Laufzeitproblemen führt.

Siehe auch *Dateisystemberechtigungen* und *Hintergrundaufgaben mit Celery*.

Webserver ausführen

Die Ausführung von Weblate unterscheidet sich nicht von der Ausführung anderer Django-basierter Programme. Django wird normalerweise als uWSGI oder fcgi ausgeführt (siehe Beispiele für verschiedene Webserver unten).

Zu Testzwecken können Sie den in Django integrierten Webserver verwenden:

```
weblate runserver
```

Warnung: VERWENDEN SIE DIESEN SERVER NICHT IN EINER PRODUKTIONSUMGEBUNG. Er hat keine Sicherheits- oder Leistungstests durchlaufen. Siehe auch Django-Dokumentation zu `runserver`.

Hinweis: Der integrierte Django-Server stellt statische Dateien nur mit aktiviertem `DEBUG` bereit, da er nur für die Entwicklung gedacht ist. Für den produktiven Einsatz siehe die wsgi-Einstellungen in *Beispielkonfiguration für*

NGINX und uWSGI, Beispielkonfiguration für Apache, Beispielkonfiguration für Apache und Gunicorn und Bereitstellung statischer Dateien.

Bereitstellung statischer Dateien

Geändert in Version 2.4: Vor der Version 2.4 nutzte Weblate das Django-Framework für statische Dateien nicht richtig und die Einrichtung war komplexer.

Django muss seine statischen Dateien in einem einzigen Verzeichnis sammeln. Um dies zu tun, führen Sie `weblate collectstatic --noinput` aus. Dadurch werden die statischen Dateien in ein Verzeichnis kopiert, das durch die Einstellung `STATIC_ROOT` festgelegt ist (dies ist standardmäßig ein `static` Verzeichnis innerhalb von `DATA_DIR`).

Es wird empfohlen, statische Dateien direkt von Ihrem Webserver bereitzustellen, was Sie für die folgenden Pfade verwenden sollten:

/static/

Stellt statische Dateien für Weblate und das Admin-Interface bereit (definiert durch `STATIC_ROOT`).

/media/

Wird für Medien-Uploads durch Benutzer (z. B. Bildschirmfotos) verwendet.

/favicon.ico

Sollte umgeschrieben werden, um eine Regel zur Bereitstellung von `/static/favicon.ico` umzuschreiben.

Siehe auch:

Beispielkonfiguration für NGINX und uWSGI, Beispielkonfiguration für Apache, Beispielkonfiguration für Apache und Gunicorn, Client-Assets komprimieren, How to deploy Django, How to deploy static files

Richtlinie zur Inhaltssicherheit

Die Standardkonfiguration von Weblate aktiviert die Middleware `weblate.middleware.SecurityMiddleware`, die sicherheitsbezogene HTTP-Header wie `Content-Security-Policy` oder `X-XSS-Protection` setzt. Diese sind standardmäßig so eingerichtet, dass sie mit Weblate und seiner Konfiguration zusammenarbeiten, aber dies muss möglicherweise an Ihre Umgebung angepasst werden.

Siehe auch:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

Beispielkonfiguration für NGINX und uWSGI

Um einen produktiven Webserver zu betreiben, verwenden Sie den mit Weblate installierten `wsgi`-Wrapper (in der virtuellen Umgebung wird er als `~/weblate-env/lib/python3.9/site-packages/weblate/wsgi.py` installiert). Vergessen Sie nicht, auch den Python-Suchpfad in Ihrer virtuellen Umgebung zu setzen (zum Beispiel mit `virtualenv = /home/user/weblate-env` in `uWSGI`).

In der folgenden Konfiguration wird Weblate als `uWSGI` unter dem `NGINX`-Webserver ausgeführt.

Konfiguration für `NGINX` (auch verfügbar als `weblate/examples/weblate.nginx.conf`):

```
#
# nginx configuration for Weblate
#
# You will want to change:
#
# - server_name
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
server {
    listen 80;
    server_name weblate;
    # Not used
    root /var/www/html;

    location ~ ^/favicon.ico$ {
        # DATA_DIR/static/favicon.ico
        alias /home/weblate/data/static/favicon.ico;
        expires 30d;
    }

    location /static/ {
        # DATA_DIR/static/
        alias /home/weblate/data/static/;
        expires 30d;
    }

    location /media/ {
        # DATA_DIR/media/
        alias /home/weblate/data/media/;
        expires 30d;
    }

    location / {
        include uwsgi_params;
        # Needed for long running operations in admin interface
        uwsgi_read_timeout 3600;
        # Adjust based to uwsgi configuration:
        uwsgi_pass unix:///run/uwsgi/app/weblate/socket;
        # uwsgi_pass 127.0.0.1:8080;
    }
}
```

Konfiguration für uWSGI (auch verfügbar als `weblate/examples/weblate.uwsgi.ini`):

```
#
# uWSGI configuration for Weblate
#
# You will want to change:
#
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
[uwsgi]
plugins      = python3
master       = true
protocol     = uwsgi
socket       = 127.0.0.1:8080
wsgi-file    = /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/wsgi.
↳py

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path = /path/to/weblate
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# In case you're using virtualenv uncomment this:
virtualenv = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size = 8192

# Reload when consuming too much of memory
reload-on-rss = 250

# Increase number of workers for heavily loaded sites
workers = 8

# Enable threads for Sentry error submission
enable-threads = true

# Child processes do not need file descriptors
close-on-exec = true

# Avoid default 0000 umask
umask = 0022

# Run as weblate user
uid = weblate
gid = weblate

# Enable harakiri mode (kill requests after some time)
# harakiri = 3600
# harakiri-verbose = true

# Enable uWSGI stats server
# stats = :1717
# stats-http = true

# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

Siehe auch:

How to use Django with uWSGI

Beispielkonfiguration für Apache

Es wird empfohlen, prefork MPM zu verwenden, wenn WSGI mit Weblate genutzt wird.

Die folgende Konfiguration lässt Weblate als WSGI laufen, Sie müssen `mod_wsgi` aktiviert haben (verfügbar als `weblate/examples/apache.conf`):

```
#
# VirtualHost for Weblate
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate_
↪request-timeout=600
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.9/site-packages/
↪weblate/wsgi.py process-group=weblate
    WSGIPassAuthorization On

    <Directory /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/>
        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>

</VirtualHost>
```

Bemerkung: Weblate erfordert Python 3, also stellen Sie bitte sicher, dass Sie die Python 3-Variante des modwsgi verwenden. Normalerweise ist es als separates Paket verfügbar, zum Beispiel `libapache2-mod-wsgi-py3`.

Siehe auch:

System-Sprachumgebungen und -Kodierung, *How to use Django with Apache and mod_wsgi*

Beispielkonfiguration für Apache und Gunicorn

Die folgende Konfiguration führt Weblate in Gunicorn und Apache 2.4 aus (verfügbar als `weblate/examples/apache.gunicorn.conf`):

```
#
# VirtualHost for Weblate using gunicorn on localhost:8000
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:443>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/https_cert.cert
    SSLCertificateKeyFile /etc/apache2/ssl/https_key.pem
    SSLProxyEngine On

    ProxyPass /favicon.ico !
    ProxyPass /static/ !
    ProxyPass /media/ !

    ProxyPass / http://localhost:8000/
    ProxyPassReverse / http://localhost:8000/
    ProxyPreserveHost On
</VirtualHost>
```

Siehe auch:[How to use Django with Gunicorn](#)**Weblate unter Pfad ausführen**

Neu in Version 1.3.

Es wird empfohlen, prefork MPM zu verwenden, wenn WSGI mit Weblate genutzt wird.

Eine beispielhafte Apache-Konfiguration für die Bereitstellung von Weblate unter `/weblate`. Wiederum unter `mod_wsgi` (auch verfügbar als `weblate/examples/apache-path.conf`):

```
#
# VirtualHost for Weblate, running under /weblate path
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:80>
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

ServerAdmin admin@weblate.example.org
ServerName weblate.example.org

# DATA_DIR/static/favicon.ico
Alias /weblate/favicon.ico /home/weblate/data/static/favicon.ico

# DATA_DIR/static/
Alias /weblate/static/ /home/weblate/data/static/
<Directory /home/weblate/data/static/>
    Require all granted
</Directory>

# DATA_DIR/media/
Alias /weblate/media/ /home/weblate/data/media/
<Directory /home/weblate/data/media/>
    Require all granted
</Directory>

# Path to your Weblate virtualenv
WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate_
↪request-timeout=600
WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.9/site-packages/
↪weblate/wsgi.py process-group=weblate
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/>
    <Files wsgi.py>
        Require all granted
    </Files>
</Directory>

</VirtualHost>

```

Zusätzlich müssen Sie `weblate/settings.py` anpassen:

```
URL_PREFIX = "/weblate"
```

2.1.11 Hintergrundaufgaben mit Celery

Neu in Version 3.2.

Weblate verwendet Celery, um regelmäßige Aufgaben und Hintergrundaufgaben auszuführen. Sie sollten einen Celery-Dienst laufen lassen, der diese ausführt. Er ist zum Beispiel für die folgenden Vorgänge zuständig (diese Liste ist nicht vollständig):

- Empfang von Webhooks von externen Diensten (siehe *Benachrichtigungs-Hooks*).
- Durchführung regelmäßiger Wartungsaufgaben wie Sicherungen, Bereinigungen, tägliche Erweiterungen oder Aktualisierungen (siehe *Sichern und Verschieben von Weblate*, `BACKGROUND_TASKS`, *Erweiterungen*).
- Ausführung von *Automatische Übersetzung*.
- Zusammenfassungen-Benachrichtigungen senden.
- Auslagerung kostspieliger Operationen aus dem wsgi-Prozess.
- Übergabe ausstehender Änderungen (siehe *Lazy Commits*).

Eine typische Einrichtung mit Redis als Backend sieht wie folgt aus:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Siehe auch:

Redis-Broker-Konfiguration in Celery

Sie sollten auch den Celery-Worker starten, um die Aufgaben zu verarbeiten und geplante Aufgaben zu starten. Dies kann direkt auf der Befehlszeile erfolgen (was vor allem beim Debuggen oder Entwickeln nützlich ist):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Bemerkung: Der Celery-Prozess muss unter demselben Benutzer wie der WSGI-Prozess ausgeführt werden, sonst werden die Dateien im `DATA_DIR` mit unterschiedlichen Besitzverhältnissen gespeichert, was zu Laufzeitproblemen führt.

Siehe auch *Dateisystemberechtigungen* und *Hintergrundaufgaben mit Celery*.

Ausführung von Celery-Aufgaben in der wsgi unter Verwendung des Eager-Modus

Bemerkung: Dies hat schwerwiegende Auswirkungen auf die Leistung der Weboberfläche und beeinträchtigt Funktionen, die von regelmäßigen Auslösern abhängen (z. B. das Übertragen ausstehender Änderungen, Zusammenfassungen-Benachrichtigungen oder Backups).

Für Entwicklungszwecke möchten Sie vielleicht eine Eager-Konfiguration verwenden, die alle Aufgaben an Ort und Stelle verarbeitet:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Celery als Systemdienst ausführen

Höchstwahrscheinlich werden Sie Celery als Daemon laufen lassen wollen, und das wird in [Daemonization](#) behandelt. Für die gebräuchlichste Linux-Konfiguration mit systemd können Sie die Beispieldateien verwenden, die, wie im Ordner `examples` unten aufgelistet, geliefert werden.

Die Systemd-Unit muss als `/etc/systemd/system/celery-weblate.service` abgelegt werden:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target

[Service]
Type=forking
User=weblate
Group=weblate
EnvironmentFile=/etc/default/celery-weblate
WorkingDirectory=/home/weblate
RuntimeDirectory=celery
RuntimeDirectoryPreserve=restart
LogsDirectory=celery
ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} \
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
-A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
--logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES} \
--pidfile=${CELERYD_PID_FILE}'
ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} \
-A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
--logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
```

[Install]

```
WantedBy=multi-user.target
```

Umgebungskonfiguration als `:file:/etc/default/celery-weblate` abzulegen:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-multiplier:celery=4 \
--queues:notify=notify --prefetch-multiplier:notify=10 \
--queues:memory=memory --prefetch-multiplier:memory=10 \
--queues:translate=translate --prefetch-multiplier:translate=4 \
--concurrency:backup=1 --queues:backup=backup --prefetch-multiplier:backup=2"

# Logging configuration
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
# and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"
```

Zusätzliche Konfiguration für die Rotation der Celery-Logs mit **logrotate**, die als `/etc/logrotate.d/celery` abgelegt wird:

```
/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}
```

Regelmäßige Aufgaben mit Celery Beat

Weblate wird mit integrierten Einstellungen für geplante Aufgaben geliefert. Sie können jedoch zusätzliche Aufgaben in `settings.py` definieren, siehe zum Beispiel *Lazy Commits*.

Die Aufgaben sollen vom Celery Beats Daemon ausgeführt werden. Wenn er nicht richtig funktioniert, läuft er möglicherweise nicht oder seine Datenbank wurde beschädigt. Überprüfen Sie in einem solchen Fall die Startprotokolle von Celery, um die Ursache herauszufinden.

Celery-Status überwachen

Sie können die aktuelle Länge der Celery-Aufgabenwarteschlangen im *Verwaltungsoberfläche* finden oder Sie können `celery_queues` auf der Kommandozeile verwenden. Falls die Warteschlange zu lang wird, erhalten Sie auch einen Konfigurationsfehler in der Verwaltungsoberfläche.

Warnung: Die Celery-Fehler werden standardmäßig nur im Celery-Log protokolliert und sind für den Benutzer nicht sichtbar. Falls Sie einen Überblick über solche Fehler haben möchten, wird empfohlen, *Sammeln von Fehlerberichten* zu konfigurieren.

Siehe auch:

Weblate überwachen, *Wie überprüft man ob Weblate richtig eingerichtet ist?*, *Configuration and defaults*, *Workers Guide*, *Daemonization*, *Monitoring and Management Guide*, `celery_queues`

2.1.12 Weblate überwachen

Weblate stellt die URL `/healthz/` zur Verfügung, die für einfache Gesundheitsprüfungen verwendet werden kann, zum Beispiel mit Kubernetes. Der Docker-Container verfügt über integrierte Gesundheitsprüfungen, die diese URL verwenden.

Zur Überwachung der Metriken von Weblate können Sie den `GET /api/metrics/` API-Endpunkt verwenden.

Siehe auch:

Wie überprüft man ob Weblate richtig eingerichtet ist?, *Celery-Status überwachen*, Weblate-Plugin für Munin <<https://github.com/WeblateOrg/munin>>`_

2.1.13 Sammeln von Fehlerberichten

Weblate kann, wie jede andere Software auch, ausfallen. Um nützliche Fehlermeldungen zu sammeln, empfehlen wir die Nutzung von Diensten Dritter, um solche Informationen zu sammeln. Dies ist besonders nützlich bei fehlgeschlagenen Celery-Aufgaben, die sonst nur Fehler in den Logs melden würden und über die Sie nicht benachrichtigt werden. Weblate bietet Unterstützung für die folgenden Dienste:

Sentry

Weblate hat integrierte Unterstützung für *Sentry*. Um sie zu nutzen, genügt es, `SENTRY_DSN` in der `settings.py` zu setzen:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate hat integrierte Unterstützung für [Rollbar](#). Um sie zu nutzen, genügt es, den Anweisungen für [Rollbar-Notifier für Python](#) zu folgen.

Kurz gesagt, Sie müssen `settings.py` anpassen:

```
# Add rollbar as last middleware:
MIDDLEWARE = [
    # ... other middleware classes ...
    "rollbar.contrib.django.middleware.RollbarNotifierMiddleware",
]

# Configure client access
ROLLBAR = {
    "access_token": "POST_SERVER_ITEM_ACCESS_TOKEN",
    "client_token": "POST_CLIENT_ITEM_ACCESS_TOKEN",
    "environment": "development" if DEBUG else "production",
    "branch": "main",
    "root": "/absolute/path/to/code/root",
}
```

Alles andere wird automatisch integriert, Sie werden jetzt sowohl server- als auch clientseitige Fehler sammeln.

2.1.14 Migration von Weblate auf einen anderen Server

Die Migration von Weblate auf einen anderen Server sollte ziemlich einfach sein, es speichert jedoch Daten an wenigen Orten, die Sie sorgfältig migrieren sollten. Der beste Ansatz ist, Weblate für die Migration zu stoppen.

Migration der Datenbank

Abhängig von Ihrem Datenbank-Backend haben Sie mehrere Möglichkeiten, die Datenbank zu migrieren. Der einfachste Ansatz ist die Verwendung datenbankeigener Tools, da diese normalerweise am effektivsten sind (z. B. **mysqldump** oder **pg_dump**). Alternativ können Sie auch die Replikation verwenden, falls Ihre Datenbank dies unterstützt.

Siehe auch:

Migration zwischen Datenbanken, beschrieben in [Migration von anderen Datenbanken zu PostgreSQL](#).

Migration der VCS-Repositorys

Die VCS-Repositorys, die unter `DATA_DIR` gespeichert sind, müssen ebenfalls migriert werden. Sie können sie einfach kopieren oder **rsync** verwenden, um die Migration effektiver durchzuführen.

Sonstige Anmerkungen

Vergessen Sie nicht, andere Dienste zu verschieben, die Weblate möglicherweise verwendet hat, wie Redis, Cron-Jobs oder benutzerdefinierte Authentifizierungs-Backends.

2.2 Weblate-Bereitstellungen

Weblate lässt sich einfach in Ihrer Cloud installieren. Hier finden Sie eine detaillierte Anleitung für Ihre Plattform:

- *Installation über Docker*
- *Installation auf OpenShift*
- *Unter Kubernetes installieren*

2.2.1 Drittanbieterbereitstellungen für Weblate

Bemerkung: Die folgenden Bereitstellungen werden nicht vom Weblate-Team entwickelt oder unterstützt. Teile der Einrichtung können von der Beschreibung in dieser Dokumentation abweichen.

Bitnami Weblate-Stack

Bitnami bietet einen Weblate-Stack für viele Plattformen unter [<https://bitnami.com/stack/weblate>](https://bitnami.com/stack/weblate).

Siehe auch:

Weblate packaged by Bitnami

Weblate Cloudron-Paket

Cloudron ist eine Plattform für das Selbsthosten von Webanwendungen. Weblate, das mit Cloudron installiert wird, wird automatisch auf dem neuesten Stand gehalten. Das Paket wird vom Cloudron-Team in seinem [Weblate-Paket-Repository](#) gepflegt.



Weblate in YunoHost

Das Self-Hosting-Projekt [YunoHost](#) bietet ein Paket für Weblate. Sobald Sie Ihre YunoHost-Installation haben, können Sie Weblate wie jede andere Anwendung installieren. Sie erhalten damit einen voll funktionsfähigen Stack mit Backup und Wiederherstellung, müssen aber möglicherweise noch Ihre Einstellungsdatei für bestimmte Verwendungszwecke bearbeiten.

Sie können Ihre Verwaltungsschnittstelle oder diese Schaltfläche verwenden (sie bringt Sie zu Ihrem Server):



Es ist auch möglich, die Befehlszeilenschnittstelle zu verwenden:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Upgrade von Weblate

2.3.1 Docker-Image-Upgrades

The official Docker image (see [Installation über Docker](#)) has all Weblate upgrade steps integrated. There are typically no manual steps needed besides pulling latest version.

Siehe auch:

[Aktualisieren des Docker-Containers](#)

2.3.2 Allgemeine Upgrade-Anweisungen

Before upgrading, please check the current [Software-Anforderungen](#) as they might have changed. Once all requirements are installed or updated, please adjust your `settings.py` to match changes in the configuration (consult `settings_example.py` for correct values).

Always check [Versionsspezifische Anweisungen](#) before upgrade. In case you are skipping some versions, please follow instructions for all versions you are skipping in the upgrade. Sometimes it's better to upgrade to some intermediate version to ensure a smooth migration. Upgrading across multiple releases should work, but is not as well tested as single version upgrades.

Bemerkung: Es wird empfohlen, vor dem Upgrade eine vollständige Datenbanksicherung durchzuführen, damit Sie die Datenbank zurücksetzen können, falls das Upgrade fehlschlägt, siehe [Sichern und Verschieben von Weblate](#).

1. Stop wsgi and Celery processes. The upgrade can perform incompatible changes in the database, so it is always safer to avoid old processes running while upgrading.
2. Upgrade Weblate code.

For pip installs it can be achieved by:

```
pip install -U "Weblate[all]==version"
```

Oder wenn Sie einfach nur die neueste Version erhalten möchten:

```
pip install -U "Weblate[all]"
```

If you don't want to install all of the optional dependencies do:

```
pip install -U Weblate
```

With Git checkout you need to fetch new source code and update your installation:

```
cd weblate-src
git pull
# Update Weblate inside your virtualenv
. ~/weblate-env/bin/pip install -e .
# Install dependencies directly when not using virtualenv
pip install --upgrade -r requirements.txt
# Install optional dependencies directly when not using virtualenv
pip install --upgrade -r requirements-optional.txt
```

3. New Weblate release might have new [Optionale Abhängigkeiten](#), please check if they cover features you want.
4. Upgrade configuration file, refer to `settings_example.py` or [Versionsspezifische Anweisungen](#) for needed steps.
5. Upgrade database structure:

```
weblate migrate --noinput
```

6. Collect updated static files (see *Laufender Server* and *Bereitstellung statischer Dateien*):

```
weblate collectstatic --noinput --clear
```

7. Compress JavaScript and CSS files (optional, see *Client-Assets komprimieren*):

```
weblate compress
```

8. If you are running version from Git, you should also regenerate locale files every time you are upgrading. You can do this by invoking:

```
weblate compilemessages
```

9. Verify that your setup is sane (see also *Produktionseinrichtung*):

```
weblate check --deploy
```

10. Celery-Worker neu starten (siehe *Hintergrundaufgaben mit Celery*).

2.3.3 Versionsspezifische Anweisungen

Upgrade von 2.x

Wenn Sie von der Version 2.x upgraden, sollten Sie immer zuerst auf 3.0.1 upgraden und dann mit der 3.x-Serie fortfahren. Upgrades, die diesen Schritt überspringen, werden nicht unterstützt und führen zu Fehlern.

Siehe auch:

[Upgrade von 2.20 auf 3.0 in der Weblate-3.0-Dokumentation](#)

Upgrade von 3.x

Wenn Sie von der Version 3.x aktualisieren, führen Sie immer zuerst ein Upgrade auf 4.0.4 oder 4.1.1 durch und setzen Sie dann das Upgrade in der 4.x-Serie fort. Upgrades, die diesen Schritt überspringen, werden nicht unterstützt und führen zu Fehlern.

Siehe auch:

[Upgrade von 3.11 auf 4.0 in der Weblate-4.0-Dokumentation](https://docs.weblate.org/en/weblate-4.0.4/admin/upgrade.html#upgrade-from-3-11-to-4-0) <<https://docs.weblate.org/en/weblate-4.0.4/admin/upgrade.html#upgrade-from-3-11-to-4-0>>`_

Upgrade von 4.0 auf 4.1

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

Notable configuration or dependencies changes:

- There are several changes in `settings_example.py`, most notable middleware changes, please adjust your settings accordingly.
- There are new file formats, you might want to include them in case you modified the `WEBLATE_FORMATS`.
- There are new quality checks, you might want to include them in case you modified the `CHECK_LIST`.
- There is change in `DEFAULT_THROTTLE_CLASSES` setting to allow reporting of rate limiting in the API.
- Es gibt einige neue und aktualisierte Anforderungen.
- There is a change in `INSTALLED_APPS`.

- Die Einstellung `MT_DEEPL_API_VERSION` wurde in Version 4.7 entfernt. Die maschineller Übersetzung *DeepL* verwendet nun stattdessen `MT_DEEPL_API_URL`. Möglicherweise müssen Sie `MT_DEEPL_API_URL` an Ihre Abonnements anpassen.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.1 auf 4.2

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

Notable configuration or dependencies changes:

- Upgrades von 3.x Versionen werden nicht mehr unterstützt, bitte aktualisieren Sie zuerst auf 4.0 oder 4.1.
- Es gibt einige neue und aktualisierte Anforderungen.
- There are several changes in `settings_example.py`, most notable new middleware and changed application ordering.
- The keys for JSON based formats no longer include leading dot. The strings are adjusted during the database migration, but external components might need adjustment in case you rely on keys in exports or API.
- The Celery configuration was changed to no longer use `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.
- The Weblate domain is now configured in the settings, see `SITE_DOMAIN` (or `WEBLATE_SITE_DOMAIN`). You will have to configure it before running Weblate.
- The username and email fields on user database now should be case insensitive unique. It was mistakenly not enforced with PostgreSQL.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.2 auf 4.3

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

Notable configuration or dependencies changes:

- There are some changes in quality checks, you might want to include them in case you modified the `CHECK_LIST`.
- The source language attribute was moved from project to a component what is exposed in the API. You will need to update *Weblate-Client* in case you are using it.
- The database migration to 4.3 might take long depending on number of strings you are translating (expect around one hour of migration time per 100,000 source strings).
- There is a change in `INSTALLED_APPS`.
- There is a new setting `SESSION_COOKIE_AGE_AUTHENTICATED` which complements `SESSION_COOKIE_AGE`.
- In case you were using **hub** or **lab** to integrate with GitHub or GitLab, you will need to reconfigure this, see `GITHUB_CREDENTIALS` and `GITLAB_CREDENTIALS`.

Geändert in Version 4.3.1:

- The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

Geändert in Version 4.3.2:

- The `post_update` method of add-ons now takes extra `skip_push` parameter.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.3 auf 4.4

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

Notable configuration or dependencies changes:

- There is a change in `INSTALLED_APPS`, `weblate.configuration` has to be added there.
- Django 3.1 ist jetzt erforderlich.
- In case you are using MySQL or MariaDB, the minimal required versions have increased, see *MySQL und MariaDB*.

Geändert in Version 4.4.1:

- *Monolingual gettext* now uses both `msgid` and `msgctxt` when present. This will change identification of translation strings in such files breaking links to Weblate extended data such as screenshots or review states. Please make sure you commit pending changes in such files prior upgrading and it is recommended to force loading of affected component using `loadpo`.
- Increased minimal required version of translate-toolkit to address several file format issues.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.4 auf 4.5

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

Notable configuration or dependencies changes:

- Bei umfangreichen Glossaren kann die Migration viel Zeit in Anspruch nehmen.
- Glossare werden jetzt als reguläre Komponenten gespeichert.
- Die Glossar-API wurde entfernt. Verwenden Sie die reguläre Übersetzungs-API für den Zugriff auf Glossare.
- There is a change in `INSTALLED_APPS` - `weblate.metrics` should be added.

Geändert in Version 4.5.1:

- Es gibt eine neue Abhängigkeit vom Modul *pyahocorasick*.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.5 auf 4.6

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

Notable configuration or dependencies changes:

- There are new file formats, you might want to include them in case you modified the `WEBLATE_FORMATS`.
- API for creating components now automatically uses *Weblate-interne URLs*, see `POST /api/projects/(string:project)/components/`.
- Es gibt eine Änderung in den Abhängigkeiten und `PASSWORD_HASHERS` um Argon2 für das Hashing von Passwörtern zu bevorzugen.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.6 auf 4.7

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

Notable configuration or dependencies changes:

- Es gibt mehrere Änderungen in `settings_example.py`, vor allem in der Middleware (`MIDDLEWARE`), bitte passen Sie Ihre Einstellungen entsprechend an.
- Die maschinelle Übersetzung deepl hat jetzt eine allgemeine Einstellung `MT_DEEPL_API_URL`, um sich flexibler an verschiedene Abonnementmodelle anzupassen. Die Einstellung `MT_DEEPL_API_VERSION` wird nicht mehr verwendet.
- Django 3.2 ist jetzt erforderlich.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.7 auf 4.8

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

In dieser Version sind keine zusätzlichen Upgrade-Schritte erforderlich.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.8 auf 4.9

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- There is a change in storing metrics, the upgrade can take long time on larger sites.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.0 auf 4.10

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- There is a change in per-project groups, the upgrade can take long time on sites with thousands of projects.
- Django 4.0 has made some incompatible changes, see [Backwards incompatible changes in 4.0](#). Weblate still supports Django 3.2 for now, in case any of these are problematic. Most notable changes which might affect Weblate:
 - Die Unterstützung für PostgreSQL 9.6 eingestellt, Django 4.0 unterstützt PostgreSQL 10 und höher.
 - Das Format von `CSRF_TRUSTED_ORIGINS` wurde geändert.
- Der Docker-Container verwendet jetzt Django 4.0, siehe oben für Änderungen.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.10 auf 4.11

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- Weblate erfordert jetzt Python 3.7 oder neuer.
- Die Implementierung von *Verwaltung der projektspezifischen Zugriffssteuerung* wurde geändert, indem das Projektpräfix aus den Gruppennamen entfernt wurde. Dies betrifft API-Benutzer.
- Weblate verwendet jetzt das Modul `charset-normalizer` anstelle von `chardet` für die Zeichensatzerkennung.
- **Geändert in 4.11.1:** Es gibt eine Änderung in `REST_FRAMEWORK` (Entfernung eines der Backends in `DEFAULT_AUTHENTICATION_CLASSES`).

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.11 auf 4.12

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- Es sind keine besonderen Schritte erforderlich.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.12 auf 4.13

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- Die *Sprachdefinitionen* werden nun automatisch bei einem Upgrade aktualisiert, verwenden Sie `UP-DATE_LANGUAGES` um dies zu deaktivieren.
- Handling of context and location has been changed for *Windows RC-Dateien*, *HTML-Dateien*, *IDML-Format*, and *Textdateien* file formats. In most cases the context is now shown as location.
- Die maschinellen Übersetzungsdienste werden nun über die Benutzeroberfläche konfiguriert, die Einstellungen aus der Konfigurationsdatei werden bei der Datenbankmigration importiert.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.13 auf 4.14

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- Die Java-Formatierungsprüfungen entsprechen nun den GNU-gettext-Markierungen. Die in Weblate gesetzten Markierungen werden automatisch migriert, aber Skripte von Drittanbietern müssen `java-printf-format` anstelle von `java-format` und `java-format` anstelle von `java-messageformat` verwenden.
- Die *jellyfish*-Abhängigkeit wurde durch *rapidfuzz* ersetzt.
- **Changed in 4.14.2:** Deprecated insecure configuration of VCS service API keys via `_TOKEN/_USERNAME` configuration instead of `_CREDENTIALS` list. In Docker, please add matching `_HOST` directive. For example see `WEBLATE_GITHUB_HOST` and `GITHUB_CREDENTIALS`.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.14 auf 4.15

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- Weblate benötigt jetzt die `btree_gin` Erweiterung in PostgreSQL. Der Migrationsprozess wird sie installieren, wenn er genügend Rechte hat. Siehe *Erstellen einer Datenbank in PostgreSQL* für die manuelle Einrichtung.
- Das Docker-Image aktiviert den Debugmodus nicht mehr standardmäßig. Falls Sie ihn wünschen, aktivieren Sie ihn in der Einsatzumgebung mit `WEBLATE_DEBUG`.
- Die Datenbankmigration kann bei größeren Instanzen Stunden dauern, da einige der Indizes neu erstellt werden müssen.
- **Geändert in 4.15.1:** Der Standardwert für `DEFAULT_PAGINATION_CLASS` in den Rest-Framework-Einstellungen geändert.

Siehe auch:

Allgemeine Upgrade-Anweisungen

Upgrade von 4.15 auf 4.16

Please follow *Allgemeine Upgrade-Anweisungen* in order to perform update.

- Celery beat is now storing the tasks schedule in the database, `CELERY_BEAT_SCHEDULER` and `INSTALLED_APPS` need to be changed for that.
- The deprecated VCS setting for credentials is no longer supported, see *Upgrade von 4.13 auf 4.14*.
- Upgrade of *django-crispy-forms* requires changes in `INSTALLED_APPS`.
- Integration of *django-cors-headers* requires changes in `INSTALLED_APPS` and `MIDDLEWARE`.

Siehe auch:

Allgemeine Upgrade-Anweisungen

2.3.4 Upgrade von Python 2 auf Python 3

Weblate unterstützt kein Python das älter als 3.6 ist. Falls Sie noch mit einer älteren Version arbeiten, führen Sie bitte zuerst die Migration auf Python 3 mit der bestehenden Version durch und aktualisieren Sie später. Siehe *Upgrade von Python 2 auf Python 3 in der Weblate-3.11.1-Dokumentation*.

2.3.5 Migration von anderen Datenbanken zu PostgreSQL

Wenn Sie Weblate auf einer anderen Datenbank als PostgreSQL betreiben, sollten Sie eine Migration zu PostgreSQL in Betracht ziehen, da Weblate damit am besten funktioniert. Die folgenden Schritte werden Sie bei der Migration Ihrer Daten zwischen den Datenbanken anleiten. Bitte denken Sie daran, sowohl den Web- als auch den Celery-Server vor der Migration zu stoppen, da es sonst zu inkonsistenten Daten kommen kann.

Erstellen einer Datenbank in PostgreSQL

Normalerweise ist es eine gute Idee, Weblate in einer separaten Datenbank und unter einem separaten Benutzerkonto laufen zu lassen:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate
```

(Fortsetzung auf der nächsten Seite)

```
# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Migration mit Django-JSON-Dumps

Der einfachste Ansatz für die Migration ist die Verwendung von Django-JSON-Dumps. Dies funktioniert gut für kleinere Installationen. Bei größeren Sites sollten Sie stattdessen pgloader verwenden, siehe *Migration zu PostgreSQL mit pgloader*.

1. Fügen Sie PostgreSQL als zusätzliche Datenbankverbindung in `settings.py` ein:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
        # Additional database options
        "OPTIONS": {
            # In case of using an older MySQL server, which has MyISAM as a
            ↳ default storage
            # 'init_command': 'SET storage_engine=INNODB',
            # Uncomment for MySQL older than 5.7:
            # 'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
            # If your server supports it, see the Unicode issues above
            "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away error:
            "connect_timeout": 28800,
        },
    },
    "postgresql": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    },
}
```

2. Führen Sie die Migration durch und löschen Sie alle in die Tabellen eingefügten Daten:

```
weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql
```

3. Dump der Legacy-Datenbank und Import in PostgreSQL

```
weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql
```

4. Passen Sie `DATABASES` an, um nur die PostgreSQL-Datenbank als Standard zu verwenden, entfernen Sie die Legacy-Verbindung.

Weblate sollte nun bereit sein, um von der PostgreSQL-Datenbank aus zu laufen.

Migration zu PostgreSQL mit pgloader

Der `pgloader` ist ein allgemeines Migrationswerkzeug für die Migration von Daten nach PostgreSQL. Sie können es verwenden, um Weblate-Datenbanken zu migrieren.

1. Passen Sie `settings.py` an, um PostgreSQL als Datenbank zu verwenden.
2. Migrate the schema in the PostgreSQL database:

```
weblate migrate
weblate sqlflush | weblate dbshell
```

3. Run the `pgloader` to transfer the data. The following script can be used to migrate the database, but you might want to learn more about `pgloader` to understand what it does and tweak it to match your setup:

```
LOAD DATABASE
FROM      mysql://weblate:password@localhost/weblate
INTO      postgresql://weblate:password@localhost/weblate

WITH include no drop, truncate, create no tables, create no indexes, no_
↳foreign keys, disable triggers, reset sequences, data only

ALTER SCHEMA 'weblate' RENAME TO 'public'
;
```

2.3.6 Migrating from Pootle

As Weblate was originally written as replacement from Pootle, it is supported to migrate user accounts from Pootle. You can dump the users from Pootle and import them using `importusers`.

2.4 Sichern und Verschieben von Weblate

2.4.1 Sicherungen auf Projektebene

Neu in Version 4.14.

Warnung: Die Wiederherstellung von Backups wird nur unterstützt, wenn PostgreSQL oder MariaDB 10.5+ als Datenbank verwendet wird.

Das Projekt sichert alle Übersetzungsinhalte von Weblate (Projekt, Komponenten, Übersetzungen, Zeichenkette-Kommentare, Vorschläge oder Qualitätsprüfungen). Es ist geeignet, um ein Projekt auf eine andere Weblate-Instanz zu übertragen.

Sie können eine Projektsicherung in *Verwaltung* ↓ *Sicherungen* durchführen. Die Sicherung kann beim Erstellen eines Projekts wiederhergestellt werden (siehe *Adding translation projects and components*).

Die Sicherungen enthalten derzeit keine Informationen über die Zugriffssteuerung und den Verlauf.

Die Kommentare und Vorschläge sind mit dem Benutzernamen des Benutzers, der sie erstellt hat, hinterlegt. Beim Import werden sie einem passenden Benutzer zugewiesen. Wenn es keinen Benutzer mit einem solchen Benutzernamen gibt, wird er einem anonymen Benutzer zugewiesen.

Die erzeugten Backups werden auf dem Server aufbewahrt, wie durch `PROJECT_BACKUP_KEEP_DAYS` und `PROJECT_BACKUP_KEEP_COUNT` konfiguriert (standardmäßig werden maximal 3 Backups für 30 Tage aufbewahrt).

2.4.2 Automatisierte Datensicherung mit BorgBackup


Neu in Version 3.9.

Weblate hat integrierte Unterstützung für die Erstellung von Service-Backups mit **BorgBackup**. Borg erstellt platzsparende verschlüsselte Backups, die sicher in der Cloud gespeichert werden können. Die Backups können in der Verwaltungsoberfläche über den Reiter *Sicherungen* gesteuert werden.

Geändert in Version 4.4.1: Sowohl PostgreSQL- als auch MySQL/MariaDB-Datenbanken sind in den automatischen Backups enthalten.

Die Backups mit Borg sind inkrementell, und Weblate ist so konfiguriert, dass die folgenden Backups beibehalten werden:

- Tägliche Backups der letzten 14 Tage
- Wöchentliche Backups der letzten 8 Wochen
- Monatliche Backups der letzten 6 Monate

 Weblate
 Dashboard Projects Languages Checks

Manage / Backups

Backup process triggered

Weblate status
 Backups
 Translation memory
 Performance report
 SSH keys
 Alerts
 Repositories
 Users
 Teams

Appearance
 Tools
 Automatic suggestions
 Billing

Backup service: /tmp/tmp8o1nk_tnweblate

Backup service credentials March 1, 2023

Backup repository /tmp/tmp8o1nk_tnweblate

 Passphrase 2zwuVXXKVI5dr\$pyvuga fL6FwBdzr6CYBTUXwT\$z^OmH0&gc^ (

 The passphrase is used to encrypt the backups and is necessary to restore them.

 SSH key

 Download private key

 The private key is needed to access the remote backup repository.

Deleted the oldest backups March 1, 2023

Backup performed March 1, 2023

Repository initialization March 1, 2023

Turn off
 Perform backup
 Delete

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

 Please enter the activation token obtained when making the subscription.

Activate
 Purchase support package

Add backup service

Backup repository URL

 Use /path/to/repo for local backups or user@host:/path/to/repo or ssh://user@host:port/path/to/backups for remote SSH backups.

Add

Powered by Weblate 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Borg-Verschlüsselungsschlüssel

BorgBackup erstellt verschlüsselte Backups, die Sie ohne die Passphrase nicht wiederherstellen können. Die Passphrase wird beim Hinzufügen eines neuen Backup-Dienstes generiert und Sie sollten sie kopieren und an einem sicheren Ort aufbewahren.

Wenn Sie *Von Weblate bereitgestellter Backup-Speicher* verwenden, sichern Sie bitte auch Ihren privaten SSH-Schlüssel, da dieser für den Zugriff auf Ihre Backups verwendet wird.

Siehe auch:

`borg init`

Backup anpassen

- Das Datenbank-Backup kann über `DATABASE_BACKUP` konfiguriert werden.
- Die Erstellung von Sicherungskopien kann mit `BORG_EXTRA_ARGS` angepasst werden.

2.4.3 Von Weblate bereitgestellter Backup-Speicher

Der einfachste Weg, Ihre Weblate-Instanz zu sichern, ist der Erwerb des **Backup-Service** unter weblate.org. So bringen Sie ihn zum Laufen:

1. Erwerben Sie den *Backup-Dienst* auf <https://weblate.org/support/#backup>.
2. Geben Sie den erhaltenen Schlüssel in die Verwaltungsoberfläche ein, siehe *Integrating support*.
3. Weblate stellt eine Verbindung zum Cloud-Dienst her und erhält die Zugangsdaten für die Sicherungen.
4. Aktivieren Sie die neue Sicherungskonfiguration auf der Reiterkarte *Sicherungen*.
5. Sichern Sie Ihre Borg-Zugangsdaten, um die Backups wiederherstellen zu können, siehe *Borg-Verschlüsselungsschlüssel*.

Hinweis: Der manuelle Schritt, alles einzuschalten, dient Ihrer Sicherheit. Ohne Ihre Zustimmung werden keine Daten an den Backup-Speicher gesendet, den Sie durch den Registrierungsprozess erhalten haben.

2.4.4 Verwendung von eigenem Backup-Speicher

Sie können auch Ihren eigenen Speicher für die Backups verwenden. SSH kann verwendet werden, um Backups im entfernten Ziel zu speichern, der Zielservers muss **BorgBackup** installiert haben.

Siehe auch:

[General](#) in der Borg-Dokumentation

Lokales Dateisystem

Es wird empfohlen, den absoluten Pfad für das lokale Backup anzugeben, zum Beispiel `/path/to/backup`. Das Verzeichnis muss für den Benutzer, unter dem Weblate läuft, beschreibbar sein (siehe *Dateisystemberechtigungen*). Existiert es nicht, versucht Weblate, es zu erstellen, benötigt dafür aber die entsprechenden Berechtigungen.

Hinweis: Wenn Sie Weblate in Docker ausführen, stellen Sie bitte sicher, dass der Speicherort des Backups vom Weblate-Container als Volume freigegeben wird. Andernfalls werden die Backups beim Neustart des Containers, in dem sie sich befinden, von Docker verworfen.

Eine Möglichkeit ist, Backups in ein bestehendes Volume zu legen, zum Beispiel `/app/data/borgbackup`. Dies ist ein vorhandenes Volume im Container.

Sie können auch einen neuen Container für die Backups in der Docker-Compose-Datei hinzufügen, indem Sie beispielsweise `/borgbackup` verwenden:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

Das Verzeichnis, in dem die Backups gespeichert werden, muss der UID 1000 gehören, ansonsten kann Weblate die Backups nicht dorthin schreiben.

Remote-Backups

Um Remote-Backups zu erstellen, müssen Sie [BorgBackup](#) auf einem anderen Server installieren, der für Ihre Weblate-Installation über SSH mit dem Weblate-SSH-Schlüssel erreichbar ist:

1. Bereiten Sie einen Server vor, auf dem Ihre Backups gespeichert werden sollen.
2. Installieren Sie den SSH-Server darauf (bei den meisten Linux-Distributionen erhalten Sie ihn standardmäßig).
3. Installieren Sie *BorgBackup* auf diesem Server; für die meisten Linux-Distributionen sind Pakete verfügbar (siehe [Installation](#)).
4. Wählen Sie einen vorhandenen Benutzer oder erstellen Sie einen neuen Benutzer, der für die Sicherung verwendet werden soll.
5. Fügen Sie dem Benutzer den SSH-Schlüssel von Weblate hinzu, damit Weblate ohne Passwort per SSH auf den Server zugreifen kann (siehe [Weblate-SSH-Schlüssel](#)).
6. Konfigurieren Sie den Backup-Speicherort in Weblate als `user@host:/path/to/backups` oder `ssh://user@host:port/path/to/backups`.

Hinweis: *Von Weblate bereitgestellter Backup-Speicher* bietet Ihnen automatisierte Remote-Backups ohne jeglichen Aufwand.

Siehe auch:

[Weblate-SSH-Schlüssel](#), [General](#)

2.4.5 Wiederherstellung aus BorgBackup

1. Stellen Sie den Zugriff auf Ihr Backup-Repository wieder her und bereiten Sie Ihre Backup-Passphrase vor.
2. Listen Sie mit `borg list REPOSITORY` alle Backups auf dem Server auf.
3. Stellen Sie die gewünschte Sicherung mit `borg extract REPOSITORY::ARCHIVE` in das aktuelle Verzeichnis wieder her.
4. Stellen Sie die Datenbank aus dem SQL-Dump wieder her, der sich im Verzeichnis `backup` im Weblate-Datenverzeichnis befindet (siehe [Gedumpte Daten für Backups](#)).
5. Kopieren Sie die Weblate-Konfiguration (`backups/settings.py`, siehe [Gedumpte Daten für Backups](#)) an den richtigen Ort, siehe [Anpassen der Konfiguration](#).

Bei der Verwendung von Docker-Containern ist die Einstellungsdatei bereits im Container enthalten und Sie sollten die ursprünglichen Umgebungsvariablen wiederherstellen. Die Datei `environment.yml` kann Ihnen dabei helfen (siehe [Gedumpte Daten für Backups](#)).

6. Kopieren Sie das gesamte wiederhergestellte Datenverzeichnis an den mit `DATA_DIR` konfigurierten Ort.

Bei der Verwendung von Docker-Containern legen Sie die Daten in das Daten-Volume, siehe [Docker-Container-Volumes](#).

Bitte vergewissern Sie sich, dass die Dateien die korrekten Besitzverhältnisse und Berechtigungen haben, siehe [Dateisystemberechtigungen](#).

Die Borg-Sitzung könnte wie folgt aussehen:

```
$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
2019-09-26T14:56:08          Thu, 2019-09-26 14:56:08
→ [de0e0f13643635d5090e9896bdaceb92a023050749ad3f3350e788f1a65576a5]
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:
```

Siehe auch:

`borg list`, `borg extract`

2.4.6 Manuelle Sicherung

Je nachdem, was Sie speichern möchten, sichern Sie die Art der Daten, die Weblate an den jeweiligen Stellen speichert.

Hinweis: Wenn Sie manuelle Backups durchführen, können Sie die Warnung von Weblate über fehlende Backups ausschalten, indem Sie `weblate.I028` zu `SILENCED_SYSTEM_CHECKS` in `settings.py` oder `WEBLATE_SILENCED_SYSTEM_CHECKS` für Docker hinzufügen.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

Datenbank

Der tatsächliche Speicherort hängt von der Einrichtung Ihrer Datenbank ab.

Hinweis: Die Datenbank ist der wichtigste Speicher. Legen Sie regelmäßig Sicherungskopien Ihrer Datenbank an. Ohne die Datenbank sind alle Übersetzungen verloren.

Native Datenbanksicherung

Es wird empfohlen, einen Dump der Datenbank mit datenbankeigenen Tools wie `pg_dump` oder `mysqldump` zu speichern. Diese Methode ist normalerweise leistungsfähiger als das Django-Backup und stellt vollständige Tabellen mit allen Daten wieder her.

Sie können dieses Backup in einer neueren Version von Weblate wiederherstellen. Es wird alle notwendigen Migrationen durchführen, wenn es in `migrate` ausgeführt wird. Bitte konsultieren Sie [Upgrade von Weblate](#) für detailliertere Informationen über das Upgrade zwischen Versionen.

Sicherung der Django-Datenbank

Alternativ können Sie Ihre Datenbank auch mit dem Befehl `dumpdata` von Django sichern. Auf diese Weise ist die Sicherung datenbankunabhängig und kann verwendet werden, falls Sie das Datenbank-Backend ändern möchten.

Bevor Sie die Datenbank wiederherstellen, müssen Sie genau dieselbe Weblate-Version verwenden, mit der die Sicherung erstellt wurde. Dies ist notwendig, da sich die Datenbankstruktur zwischen den einzelnen Versionen ändert und Sie die Daten auf irgendeine Weise beschädigen würden. Nachdem Sie die gleiche Version installiert haben, führen Sie alle Datenbankmigrationen mit `migrate` durch.

Danach werden einige Einträge bereits in der Datenbank erstellt und sind auch in der Datenbanksicherung enthalten. Es wird empfohlen, solche Einträge manuell über die Verwaltungsshell zu löschen (siehe *Invoking management commands*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Dateien

Wenn Sie genügend Speicherplatz haben, sichern Sie einfach das gesamte `DATA_DIR`. Das ist eine sichere Sache, auch wenn es einige Dateien enthält, die Sie nicht wollen. In den folgenden Abschnitten wird detailliert beschrieben, was Sie sichern sollten und was Sie weglassen können.

Gedumpte Daten für Backups

Geändert in Version 4.7: Der Umgebungsdump wurde als `environment.yml` hinzugefügt, um die Wiederherstellung in den Docker-Umgebungen zu erleichtern.

Gespeichert in `DATA_DIR/backups`.

Weblate legt hier verschiedene Daten ab, und Sie können diese Dateien für vollständigere Backups einbinden. Die Dateien werden täglich aktualisiert (erfordert einen laufenden Celery-Beats-Server, siehe *Hintergrundaufgaben mit Celery*). Derzeit umfasst dies:

- Weblate-Einstellungen als `settings.py` (es gibt auch eine erweiterte Version in `settings-expanded.py`).
- Sicherung der PostgreSQL-Datenbank als `Datenbank.sql`.
- Umgebungsdump als `environment.yml`.

Die Datenbanksicherungen werden standardmäßig als reiner Text gespeichert, können aber auch komprimiert oder mit `DATABASE_BACKUP` ganz übersprungen werden.

Um die Datenbanksicherung wiederherzustellen, laden Sie sie z. B. mit Hilfe der Datenbank-Tools:

```
psql --file=database.sql weblate
```

Repositorys der Versionskontrolle

Gespeichert in `DATA_DIR / vcs`.

Die Versionsverwaltung enthält eine Kopie Ihrer Upstream-Repositorys mit Weblate-Änderungen. Wenn Sie *Bei Commit gleichzeitig Pushen* für alle Ihre Übersetzungskomponenten aktiviert haben, werden alle Weblate-Änderungen Upstream aufgenommen. Es ist nicht notwendig, die Repositorys auf der Weblate-Seite zu sichern, da sie ohne Datenverlust von den Upstream-Speicherorten erneut geklont werden können.

SSH- und GPG-Schlüssel

Gespeichert in `DATA_DIR / ssh` und `DATA_DIR / home`.

Wenn Sie SSH- oder GPG-Schlüssel verwenden, die von Weblate generiert wurden, sollten Sie von diesen Speicherorten ein Backup erstellen. Andernfalls gehen die privaten Schlüssel verloren und Sie müssen neue Schlüssel generieren.

Vom Benutzer hochgeladene Dateien

Gespeichert in `DATA_DIR / media`.

Sie sollten alle vom Benutzer hochgeladenen Dateien sichern (z. B. Bildschirmfotos).

Aufgaben von Celery

Die Celery-Aufgabenwarteschlange kann einige Informationen enthalten, wird aber normalerweise nicht für ein Backup benötigt. Sie verlieren höchstens Aktualisierungen, die noch nicht im Translation Memory verarbeitet wurden. Es wird ohnehin empfohlen, bei der Wiederherstellung eine Volltext- oder Repository-Aktualisierung durchzuführen, so dass es kein Problem ist, diese zu verlieren.

Siehe auch:

Hintergrundaufgaben mit Celery

Befehlszeile für manuelles Backup

Mit einem Cron-Job können Sie einen Bash-Befehl einrichten, der z. B. täglich ausgeführt wird:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz \
↳ backups vcs ssh home media fonts secret
```

Die Zeichenkette zwischen den Anführungszeichen nach `XZ_OPT` erlaubt es Ihnen, Ihre xz-Optionen auszuwählen, zum Beispiel die Menge an Speicher, die für die Kompression verwendet wird; siehe <https://linux.die.net/man/1/xz>

Sie können die Liste der Ordner und Dateien an Ihre Bedürfnisse anpassen. Um zu vermeiden, dass das Translation Memory (im Ordner „Backups“) gespeichert wird, können Sie diese Option verwenden:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz \
↳ backups/database.sql backups/settings.py vcs ssh home media fonts secret
```

2.4.7 Wiederherstellen der manuellen Sicherung

1. Stellen Sie alle von Ihnen gesicherten Daten wieder her.
2. Aktualisieren Sie alle Repositories mit `updategit`.

```
weblate updategit --all
```

2.4.8 Verschieben einer Weblate-Installation

Verschieben Sie Ihre Installation auf ein anderes System, indem Sie die obigen Anweisungen zum Sichern und Wiederherstellen befolgen.

Siehe auch:

Upgrade von Python 2 auf Python 3, Migration von anderen Datenbanken zu PostgreSQL

2.5 Authentifizierung

2.5.1 Benutzerregistrierung

Die Standardeinstellung für Weblate ist die Verwendung von `python-social-auth`, einem Formular auf der Website zur Registrierung neuer Benutzer. Nach der Bestätigung ihrer E-Mail kann ein neuer Benutzer einen Beitrag leisten oder sich mit einem der Dienste von Drittanbietern authentifizieren.

Sie können die Registrierung neuer Benutzer auch mit `REGISTRATION_OPEN` abschalten.

Die Authentifizierungsversuche unterliegen dem *Ratenbegrenzung*.

2.5.2 Authentifizierungs-Backends

Die eingebaute Lösung von Django wird für die Authentifizierung verwendet, einschließlich verschiedener sozialer Optionen, um dies zu tun. Wenn Sie sie verwenden, können Sie die Benutzerdatenbank anderer Django-basierter Projekte importieren (siehe *Migrating from Pootle*).

Django kann zusätzlich so eingerichtet werden, dass es sich auch gegenüber anderen Mitteln authentifiziert.

Siehe auch:

Authentifizierungseinstellungen beschreibt, wie man die Authentifizierung im offiziellen Docker-Image konfiguriert.

2.5.3 Soziale Authentifizierung

Dank [Welcome to Python Social Auth's documentation!](#) unterstützt Weblate die Authentifizierung über viele Dienste von Drittanbietern wie GitLab, Ubuntu, Fedora, etc.

Bitte lesen Sie die Dokumentation für allgemeine Konfigurationsanweisungen in [Django Framework](#).

Bemerkung: Standardmäßig verlässt sich Weblate auf die Authentifizierungsdienste von Drittanbietern, um eine geprüfte E-Mail-Adresse bereitzustellen. Wenn einige der Dienste, die Sie verwenden möchten, dies nicht unterstützen, erzwingen Sie bitte die E-Mail-Validierung auf der Weblate-Seite, indem Sie `FORCE_EMAIL_VALIDATION` für sie konfigurieren. Zum Beispiel:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Siehe auch:

[Pipeline](#)

Das Aktivieren einzelner Backends ist recht einfach: Sie müssen lediglich einen Eintrag in der Einstellung `AUTHENTICATION_BACKENDS` hinzufügen und eventuell die für eine bestimmte Authentifizierungsmethode benötigten Schlüssel hinzufügen. Bitte beachten Sie, dass einige Backends standardmäßig keine Benutzer-E-Mails zur Verfügung stellen. Sie müssen diese explizit anfordern, da Weblate sonst nicht in der Lage ist, die Beiträge der Benutzer richtig zuzuordnen.

Hinweis: Die meisten der Authentifizierungs-Backends erfordern HTTPS. Sobald HTTPS in Ihrem Webserver aktiviert ist, konfigurieren Sie Weblate bitte mit `ENABLE_HTTPS` oder durch `WEBLATE_ENABLE_HTTPS` im Docker-Container so, dass es korrekt gemeldet wird.

Siehe auch:

[Python Social Auth Backend](#)

OpenID-Authentifizierung

Für OpenID-basierte Dienste ist es normalerweise nur eine Frage der Aktivierung. Der folgende Abschnitt aktiviert die OpenID-Authentifizierung für OpenSUSE, Fedora und Ubuntu:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.suse.OpenSUSEOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Siehe auch:

[OpenID](#)

GitHub-Authentifizierung

Sie müssen eine OAuth-Anwendung auf GitHub registrieren und dann Weblate alle ihre Geheimnisse mitteilen:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

GitHub sollte so konfiguriert sein, dass die Callback-URL `https://example.com/accounts/complete/github/` lautet.

Es gibt ähnliche Authentifizierungs-Backends für GitHub for Organizations und GitHub for Teams. Ihre Einstellungen heißen `SOCIAL_AUTH_GITHUB_ORG_*` und `SOCIAL_AUTH_GITHUB_TEAM_*`, und

sie erfordern zusätzliche Einstellungen des Bereichs `SOCIAL_AUTH_GITHUB_ORG_NAME` oder `SOCIAL_AUTH_GITHUB_TEAM_ID`. Ihre Callback-URLs lauten `https://example.com/accounts/complete/github-org/` und `https://example.com/accounts/complete/github-teams/`.

Bemerkung: Die von Weblate während der Authentifizierung bereitgestellte Callback-URL enthält die konfigurierte Domäne. Falls Sie Fehlermeldungen über eine nicht übereinstimmende URL erhalten, sollten Sie dies beheben, siehe *Seitendomain richtig einstellen*.

Siehe auch:

GitHub

Bitbucket-Authentifizierung

Sie müssen eine Anwendung bei Bitbucket registrieren und dann Weblate alle ihre Geheimnisse mitteilen:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True
```

Bemerkung: Die von Weblate während der Authentifizierung bereitgestellte Callback-URL enthält die konfigurierte Domäne. Falls Sie Fehlermeldungen über eine nicht übereinstimmende URL erhalten, sollten Sie dies beheben, siehe *Seitendomain richtig einstellen*.

Siehe auch:

Bitbucket

Google OAuth 2

Um Google OAuth 2 zu verwenden, müssen Sie eine Anwendung auf <https://console.developers.google.com/> registrieren und die Google+ API aktivieren.

Die Weiterleitungs-URL lautet `https://WEBLATE_SERVER/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Bemerkung: Die von Weblate während der Authentifizierung bereitgestellte Callback-URL enthält die konfigurierte Domäne. Falls Sie Fehlermeldungen über eine nicht übereinstimmende URL erhalten, sollten Sie dies beheben, siehe *Seitendomain richtig einstellen*.

Seitendomain richtig einstellen.

Siehe auch:

[Google](#)

Facebook OAuth 2

Wie bei „OAuth 2“-Diensten üblich, müssen Sie Ihre Anwendung bei Facebook registrieren. Sobald dies geschehen ist, können Sie Weblate einrichten, um es zu nutzen:

Die Weiterleitungs-URL lautet `https://WEBLATE_SERVER/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Bemerkung: Die von Weblate während der Authentifizierung bereitgestellte Callback-URL enthält die konfigurierte Domäne. Falls Sie Fehlermeldungen über eine nicht übereinstimmende URL erhalten, sollten Sie dies beheben, siehe *Seitendomain richtig einstellen*.

Siehe auch:

[Facebook](#)

GitLab OAuth 2

Um GitLab OAuth 2 zu verwenden, müssen Sie eine Anwendung auf <https://gitlab.com/profile/applications> registrieren.

Die Weiterleitungs-URL lautet `https://WEBLATE_SERVER/accounts/complete/gitlab/` und stellen Sie sicher, dass Sie den Bereich *read_user* markieren.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]

# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Bemerkung: Die von Weblate während der Authentifizierung bereitgestellte Callback-URL enthält die konfigurierte Domäne. Falls Sie Fehlermeldungen über eine nicht übereinstimmende URL erhalten, sollten Sie dies beheben, siehe

Seitendomain richtig einstellen.

Siehe auch:

GitLab

Microsoft Azure Active Directory

Weblate kann so konfiguriert werden, dass allgemeine oder spezifische Mandanten für die Authentifizierung verwendet werden.

Die Weiterleitungs-URL lautet `https://WEBLATE_SERVER/accounts/complete/azuread-oauth2/` für allgemeine und `https://WEBLATE_SERVER/accounts/complete/azuread-tenant-oauth2/` für mandantenspezifische Authentifizierung.

```
# Azure AD common

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread.AzureADOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET = ""
```

```
# Azure AD Tenant

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread_tenant.AzureADTenantOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Bemerkung: Die von Weblate während der Authentifizierung bereitgestellte Callback-URL enthält die konfigurierte Domäne. Falls Sie Fehlermeldungen über eine nicht übereinstimmende URL erhalten, sollten Sie dies beheben, siehe *Seitendomain richtig einstellen*.

Siehe auch:

Microsoft Azure Active Directory

Slack

Um Slack OAuth 2 zu nutzen, müssen Sie eine Anwendung auf <<https://api.slack.com/apps>> registrieren.

Die Weiterleitungs-URL lautet `https://WEBLATE_SERVER/accounts/complete/slack/`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.slack.SlackOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SLACK_KEY = ""
SOCIAL_AUTH_SLACK_SECRET = ""
```

Bemerkung: Die von Weblate während der Authentifizierung bereitgestellte Callback-URL enthält die konfigurierte Domäne. Falls Sie Fehlermeldungen über eine nicht übereinstimmende URL erhalten, sollten Sie dies beheben, siehe *Seitendomain richtig einstellen*.

Siehe auch:

[Slack](#)

Überschreiben von Namen und Symbolen für Authentifizierungsmethoden

Sie können den Anzeigenamen und das Symbol der Authentifizierungsmethode überschreiben, indem Sie die Einstellungen `SOCIAL_AUTH_<NAME>_IMAGE` und `SOCIAL_AUTH_<NAME>_TITLE` verwenden. Zum Beispiel würde das Überschreiben der Benennung für Auth0 wie folgt aussehen:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Passwort-Authentifizierung deaktivieren

E-Mail- und Passwort-Authentifizierung können ausgeschaltet werden, indem man `social_core.backends.email.EmailAuth` aus `AUTHENTICATION_BACKENDS` entfernt. Behalten Sie `weblate.accounts.auth.WeblateUserBackend` dort, es wird für die Kernfunktionalität von Weblate benötigt.

Durch die Deaktivierung der E-Mail-Authentifizierung werden alle E-Mail-bezogenen Funktionen deaktiviert, z. B. die Benutzereinladung oder die Funktion zum Zurücksetzen des Passworts.

Tipp: Sie können weiterhin die Passwortauthentifizierung für die Adminoberfläche verwenden, für Benutzer, die Sie dort manuell erstellen. Navigieren Sie einfach zu `/admin/login/`.

Zum Beispiel kann die Authentifizierung nur mit dem openSUSE Open ID Provider wie folgt erreicht werden:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Passwort-Authentifizierung

Standardmäßig kommt `settings.py` mit einem vernünftigen Satz von `AUTH_PASSWORD_VALIDATORS`:

- Passwörter dürfen Ihren anderen persönlichen Daten nicht zu ähnlich sein.
- Passwörter müssen mindestens 10 Zeichen enthalten.
- Passwörter können kein häufig verwendetes Passwort sein.
- Passwörter dürfen nicht ausschließlich aus Zahlen bestehen.
- Passwörter dürfen nicht aus einem einzigen Zeichen oder nur aus Leerzeichen bestehen.
- Passwörter dürfen nicht mit einem Passwort übereinstimmen, das Sie in der Vergangenheit verwendet haben.

Sie können diese Einstellung an Ihre Passwortrichtlinien anpassen.

Zusätzlich können Sie auch `django-zxcvbn-password` installieren, das eine recht realistische Schätzung des Schwierigkeitsgrades von Passwörtern liefert und es ermöglicht, Passwörter unterhalb eines bestimmten Schwellenwertes abzulehnen.

2.5.5 SAML-Authentifizierung

Neu in Version 4.1.1.

Bitte folgen Sie den Anweisungen von Python Social Auth für die Konfiguration. Bedeutende Unterschiede:

- Weblate unterstützt einen einzelnen IDP, der `weblate` in `SOCIAL_AUTH_SAML_ENABLED_IDPS` genannt werden muss.
- Die URL der SAML-XML-Metadaten lautet `/accounts/metadata/saml/`.
- Die folgenden Einstellungen werden automatisch ausgefüllt: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_SP_PUBLIC_CERT`, `SOCIAL_AUTH_SAML_SP_PRIVATE_KEY`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Beispielkonfiguration:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SAML_SP_ENTITY_ID = f"https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbn1+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}
```

Die Standardkonfiguration extrahiert Benutzerdetails aus den folgenden Attributen; konfigurieren Sie Ihre IDP so, dass sie diese bereitstellt:

Attribut	SAML-URI-Referenz
Vollständiger Name	urn:oid:2.5.4.3
Vorname	urn:oid:2.5.4.42
Nachname	urn:oid:2.5.4.4
E-Mail	urn:oid:0.9.2342.19200300.100.1.3
Benutzername	urn:oid:0.9.2342.19200300.100.1.1

Hinweis: Das obige Beispiel und das Docker-Image definieren eine IDP namens `weblate`. Möglicherweise müssen Sie diese Zeichenkette als *Relay* in Ihrem IDP konfigurieren.

Siehe auch:

SAML in Docker konfigurieren, SAML

2.5.6 LDAP-Authentifizierung

Die LDAP-Authentifizierung lässt sich am besten mit dem Paket *django-auth-ldap* erreichen. Sie können es mit den üblichen Mitteln installieren:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

Hinweis: Dieses Paket ist im Docker-Container enthalten, siehe *Installation über Docker*.

Bemerkung: Es gibt einige Inkompatibilitäten im Python LDAP 3.1.0 Modul, die Sie möglicherweise daran hindern, diese Version zu verwenden. Wenn Sie den Fehler `AttributeError: 'module' object has no attribute '_trace_level'` erhalten, könnte ein Downgrade von `python-ldap` auf 3.0.0 helfen.

Sobald Sie das Paket installiert haben, können Sie es mit der Django-Authentifizierung verbinden:

```
# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

)

# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"

# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'

# List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
    # Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
    # 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
    "email": "mail",
}

# Hide the registration form
REGISTRATION_OPEN = False

```

Bemerkung: Sie sollten 'social_core.backends.email.EmailAuth' aus der `AUTHENTICATION_BACKENDS`-Einstellung entfernen, da Benutzer sonst ihr Passwort in Weblate setzen und sich damit authentifizieren können. Das Beibehalten von 'weblate.accounts.auth.WeblateUserBackend' wird immer noch benötigt, um Berechtigungen zu vergeben und anonyme Benutzer zu ermöglichen. Außerdem können Sie sich mit einem lokalen Administratorkonto anmelden, wenn Sie es erstellt haben (z B. mit `createadmin`).

Bindungs-Passwort verwenden

Wenn Sie keine direkte Bindung für die Authentifizierung verwenden können, müssen Sie die Suche verwenden und einen Benutzer für die Bindung für die Suche angeben. Zum Beispiel:

```

import ldap
from django_auth_ldap.config import LDAPSearch

AUTH_LDAP_BIND_DN = ""
AUTH_LDAP_BIND_PASSWORD = ""
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "ou=users,dc=example,dc=com", ldap.SCOPE_SUBTREE, "(uid=%(user)s)"
)

```

Integration von Active Directory

```
import ldap
from django_auth_ldap.config import LDAPSearch, NestedActiveDirectoryGroupType

AUTH_LDAP_BIND_DN = "CN=ldap,CN=Users,DC=example,DC=com"
AUTH_LDAP_BIND_PASSWORD = "password"

# User and group search objects and types
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "CN=Users,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s)"
)

# Make selected group a superuser in Weblate
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "OU=Groups,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True
```

Siehe auch:

Django Authentication Using LDAP, Authentication

2.5.7 CAS-Authentifizierung

Die CAS-Authentifizierung kann mit einem Paket wie *django-cas-ng* erreicht werden.

Schritt eins ist die Offenlegung des E-Mail-Feldes des Benutzers über CAS. Dies muss auf dem CAS-Server selbst konfiguriert werden und setzt voraus, dass Sie mindestens CAS v2 verwenden, da CAS v1 Attribute überhaupt nicht unterstützt.

Der zweite Schritt ist die Aktualisierung von Weblate zur Verwendung Ihres CAS-Servers und Ihrer Attribute.

Um *django-cas-ng* zu installieren:

```
pip install django-cas-ng
```

Sobald Sie das Paket installiert haben, können Sie es an das Django-Authentifizierungssystem anbinden, indem Sie die Datei `settings.py` ändern:

```
# Add CAS backed, keep the Django one if you want to be able to sign in
# even without LDAP for the admin account
AUTHENTICATION_BACKENDS = (
    "django_cas_ng.backends.CASBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# CAS server address
CAS_SERVER_URL = "https://cas.example.net/cas/"

# Add django_cas_ng somewhere in the list of INSTALLED_APPS
INSTALLED_APPS = (... , "django_cas_ng")
```

Schließlich kann ein Signal verwendet werden, um das E-Mail-Feld dem Benutzerobjekt zuzuordnen. Damit dies funktioniert, müssen Sie das Signal aus dem Paket *django-cas-ng* importieren und Ihren Code mit diesem Signal verbinden. Dies in der Einstellungsdatei zu tun, kann zu Problemen führen, daher ist es empfehlenswert, dies zu tun:

- In der Methode `django.apps.AppConfig.ready()` Ihrer App-Konfiguration
- In der Datei `urls.py` des Projekts (wenn keine Modelle vorhanden sind)

```
from django_cas_ng.signals import cas_user_authenticated
from django.dispatch import receiver

@receiver(cas_user_authenticated)
def update_user_email_address(sender, user=None, attributes=None, **kwargs):
    # If your CAS server does not always include the email attribute
    # you can wrap the next two lines of code in a try/catch block.
    user.email = attributes["email"]
    user.save()
```

Siehe auch:

[Django CAS NG](#)

2.5.8 Konfigurieren der Django-Authentifizierung von Drittanbietern

Generell sollte jedes Django-Authentifizierungs-Plugin mit Weblate funktionieren. Folgen Sie einfach den Anweisungen für das Plugin und denken Sie daran, das Weblate-Benutzer-Backend installiert zu lassen.

Siehe auch:

[LDAP-Authentifizierung](#), [CAS-Authentifizierung](#)

Typischerweise besteht die Installation aus dem Hinzufügen eines Authentifizierungs-Backends zu `AUTHENTICATION_BACKENDS` und der Installation einer Authentifizierungs-App (falls es eine gibt) in `INSTALLED_APPS`:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Zugriffssteuerung

Weblate verfügt über ein fein abgestuftes Berechtigungssystem, um Benutzerberechtigungen für die ganze Instanz oder in einem begrenztem Umfang zuzuweisen.

Geändert in Version 3.0: Vor Weblate 3.0 beruhte die Berechtigungsstruktur auf Django, ist nun aber eine speziell für Weblate erstellte. Sollten Sie eine ältere Version verwenden, konsultieren Sie bitte die Dokumentation zu jener Version, die Informationen hier gelten für sie nicht.

2.6.1 Einfache Zugriffssteuerung

Wenn Sie nicht die gesamte Weblate-Installation administrieren und nur Zugriff auf die Verwaltung bestimmter Projekte haben (wie bei [Hosted Weblate](#)), beschränken sich Ihre Möglichkeiten zur Verwaltung der Zugriffskontrolle auf folgende Einstellungen. Wenn Sie keine komplexe Einrichtung benötigen, sind diese für Sie ausreichend.

Projekt-Zugriffssteuerung

Bemerkung: Diese Funktion ist für Projekte im Freien Hosting-Angebot auf Hosted Weblate nicht verfügbar.

Sie können den Zugriff des Benutzers auf einzelne Projekte einschränken, indem Sie eine andere *Zugriffssteuerung*-Einstellung wählen. Verfügbare Optionen sind:

Öffentlich

Öffentlich sichtbar, übersetzbar für alle angemeldeten Benutzer.

Geschützt

Öffentlich sichtbar, aber nur ausgewählte Benutzer können übersetzen.

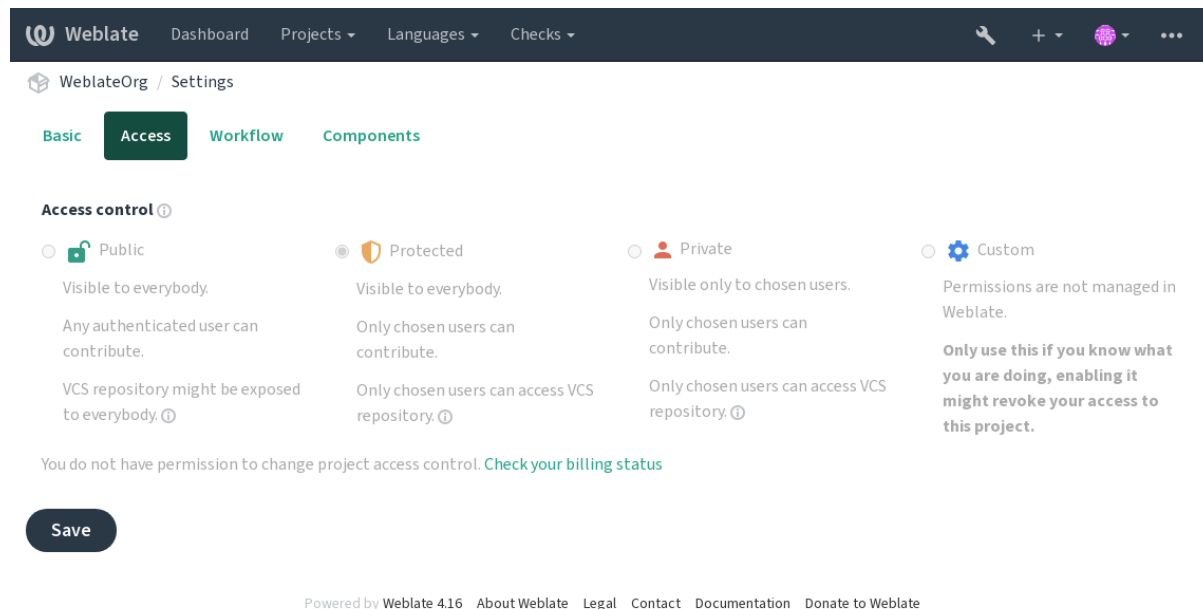
Privat

Nur ausgewählte Benutzer können es sehen und übersetzen.

Benutzerdefiniert

Die Funktionen der *Benutzerverwaltung* werden deaktiviert; standardmäßig ist es allen Benutzern untersagt, irgendwelche Aktionen am Projekt durchzuführen. Sie müssen alle Berechtigungen mit *Benutzerdefinierte Zugriffssteuerung* einrichten.

Die *Zugriffssteuerung* kann im Reiter *Zugriff* der Konfiguration (*Verwaltung* ↓ *Einstellungen*) des jeweiligen Projekts geändert werden.



Der Standardwert kann durch `DEFAULT_ACCESS_CONTROL` geändert werden.

Bemerkung: Auch bei *Privat*-Projekten werden einige Informationen über Ihr Projekt offengelegt: Statistik und Sprachzusammenfassung für die gesamte Instanz enthalten trotz der Einstellung der Zugriffskontrolle Zählungen für alle Projekte. Ihr Projektname und andere Informationen können dadurch nicht offengelegt werden.

Bemerkung: Der tatsächliche Satz an Berechtigungen, der standardmäßig für Benutzer in den Projekten *Öffentlich*, *Geschützt* und *Privat* verfügbar ist, kann vom Administrator der Weblate-Instanz mit *benutzerdefinierten Einstellungen* neu definiert werden.

Siehe auch:

Zugriffssteuerung

Verwaltung der projektspezifischen Zugriffssteuerung

Benutzer mit der Berechtigung *Projektzugriff verwalten* (siehe *Liste der Berechtigungen und integrierten Rollen*) können Benutzer in Projekten verwalten, indem sie die Benutzer zu Teams hinzufügen. Die anfängliche Sammlung von Teams wird von Weblate bereitgestellt, aber es können zusätzliche Teams definiert werden, die eine feiner abgestufte Zugriffssteuerung ermöglichen. Sie können Teams auf Sprachen beschränken und ihnen bestimmte Zugriffsrollen zuweisen (siehe *Liste der Berechtigungen und integrierten Rollen*).

Folgende Teams werden für jedes Projekt automatisch gebildet:

Für *Öffentlich*, *Geschützt* und *Privat* Projekte:

Verwaltung

Enthält alle für das Projekt verfügbaren Berechtigungen.

Überprüfung (nur wenn der *Überprüfungsablauf* eingeschaltet ist)

Darf Übersetzungen bei der Überprüfung freigeben.

Nur für *Geschützt* und *Privat* Projekte:

Übersetzen

Darf das Projekt übersetzen und offline erstellte Übersetzungen hochladen.

Quellcode

Kann Ausgangszeichenketten (wenn in den *Projekteinstellungen* erlaubt) und Infos zu Ausgangszeichenketten bearbeiten.

Sprachen

Darf übersetzte Sprachen verwalten (Übersetzungen hinzufügen oder entfernen).

Glossar

Darf das Glossar verwalten (Einträge ergänzen oder entfernen oder hochladen).

Speicher

Darf den Übersetzungsspeicher verwalten.

Bildschirmfotos

Darf Bildschirmfotos verwalten (sie hinzufügen, entfernen und mit Ausgangszeichenketten verbinden).

Automatische Übersetzung

Darf automatische Übersetzungen verwenden.

VCS

Darf das VCS verwalten und auf das exportierte Repository zugreifen.

Abrechnung

Darf auf Abrechnungsdaten und -einstellungen zugreifen (siehe Abrechnung).

Weblate

Dashboard
Projects
Languages
Checks

WeblateOrg / Access control

Users

Teams

API access

Users

Username	Full name	E-mail	Last sign in	Teams
testuser	Weblate Test	weblate@example.org	16 seconds ago	<div>Translate</div>

Once all its permissions are removed, the user will be removed from the project.

Add a user

User to add

Please type in an existing Weblate account name or e-mail address.

Add

Block user

User to block

Please type in an existing Weblate account name or e-mail address.

Block duration

Block the user until I unblock

Block

Invite new user

E-mail

Username

Username may only contain letters, numbers or the following characters:
@ . + - _

Full name

Invite

Powered by Weblate 4.16
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Diese Funktionen sind auf der Seite *Zugriffssteuerung* verfügbar, die über das Projektmenü *Verwaltung* ↓ *Benutzer* aufgerufen werden kann.

Teamadministratoren

Neu in Version 4.15.

Jedes Team kann einen Teamadministrator haben, der Benutzer innerhalb des Teams hinzufügen und entfernen kann. Dies ist nützlich, wenn Sie selbstverwaltete Teams aufbauen wollen.

Neue Benutzereinladung

Neben dem Hinzufügen eines bestehenden Benutzers zum Projekt ist es auch möglich, neue einzuladen. Jeder neue Benutzer wird sofort erstellt, aber das Konto bleibt bis zur Anmeldung mit einem Link in der Einladung, die per E-Mail gesendet wird, inaktiv. Dazu sind keine standortweiten Berechtigungen erforderlich, eine Zugriffsverwaltungs-berechtigung für den Umfang des Projekts (z. B. eine Mitgliedschaft im Team *Administration*) wäre ausreichend.

Hinweis: Wenn der eingeladene Benutzer die Gültigkeit der Einladung versäumt hat, kann er sein Passwort mit der eingeladenen E-Mail-Adresse im Formular zum Zurücksetzen des Passworts festlegen, da das Konto bereits erstellt wurde.

Neu in Version 3.11: Es ist möglich, die E-Mail für Benutzereinladungen erneut zu senden (wodurch zuvor gesendete Einladungen ungültig werden).

Die gleiche Art von Einladungen sind plattformweit über die *Verwaltungsfläche* auf der Reiterkarte *Benutzer* verfügbar.

Benutzer sperren

Neu in Version 4.7.

Für den Fall, dass sich einige Benutzer in Ihrem Projekt schlecht benehmen, haben Sie die Möglichkeit, sie für Beiträge zu sperren. Der gesperrte Benutzer kann das Projekt immer noch sehen, wenn er die entsprechenden Berechtigungen hat, aber er kann nichts dazu beitragen.

Berechtigungsverwaltung pro Projekt

Sie können Ihre Projekte auf *Geschützt* oder *Privat* einstellen und *Benutzer verwalten* pro Projekt in der Weblate-Benutzeroberfläche.

Standardmäßig verhindert dies, dass Weblate den Zugriff von *Benutzer-* und *Betrachter-Standardteams* aufgrund der eigenen Konfiguration dieser Teams gewährt. Dies hindert Sie jedoch nicht daran, die Berechtigungen für diese Projekte plattformweit zu gewähren, indem Sie die Standardteams ändern, eine neue erstellen oder zusätzliche benutzerdefinierte Einstellungen für einzelne Komponenten vornehmen, wie in *Benutzerdefinierte Zugriffssteuerung* unten beschrieben.

Einer der Hauptvorteile der Verwaltung von Berechtigungen über die Weblate-Benutzeroberfläche ist, dass Sie sie an andere Benutzer delegieren können, ohne ihnen das Superuser-Recht zu geben. Dazu fügen Sie sie dem Team *Administration* des Projekts hinzu.

2.6.2 Benutzerdefinierte Zugriffssteuerung

Bemerkung: Diese Funktion ist für Projekte im Freien Hosting-Angebot auf Hosted Weblate nicht verfügbar.

Die Berechtigungsstruktur basiert auf Teams und Rollen, wobei Rollen eine Reihe von Berechtigungen festlegen und Teams sie mit Benutzern und Übersetzungen verknüpfen, weitere Einzelheiten dazu unter *Benutzer, Rollen, Teams, und Berechtigungen*.

Die leistungsstärksten Funktionen der Zugriffssteuerung von Weblate sind bis jetzt nur über die *Django-Adminoberfläche* erreichbar. Sie können sie zur Verwaltung der Berechtigungen jedes Projekts benutzen. Sie müssen sie nicht zwingend auf *Benutzerdefinierte Zugriffssteuerung* umstellen, um sie zu verwenden. Allerdings müssen Sie Superuser-Rechte haben, um sie zu benutzen.

Wenn Sie nicht an den Details der Implementierung interessiert sind und nur eine einfache Konfiguration auf der Grundlage der Standardeinstellungen erstellen möchten oder keinen plattformweiten Zugriff auf die gesamte Weblate-Installation haben (wie bei [Hosted Weblate](#)), lesen Sie bitte den Abschnitt [Einfache Zugriffssteuerung](#).

Allgemeine Einstellungen

Dieser Abschnitt enthält einen Überblick über einige gängige Konfigurationen, die Sie interessieren könnten.

Plattformweite Rechteverwaltung

Um Berechtigungen für eine ganze Instanz auf einmal zu verwalten, fügen Sie Benutzer zu den entsprechenden [Standardteams](#) hinzu:

- *Benutzer* (dies geschieht standardmäßig durch [Automatische Teamzuweisungen](#)).
- *Prüfer* (wenn Sie den [Überprüfungsablauf](#) mit eigenen Prüfern verwenden).
- *Manager* (wenn Sie die meisten Verwaltungsaufgaben an jemand anderen delegieren wollen).

Sie sollten alle Projekte als *Public* konfigurieren (siehe [Projekt-Zugriffssteuerung](#)), da sonst die plattformweiten Berechtigungen, die durch die Mitgliedschaft in den Teams *Benutzer* und *Prüfer* gegeben sind, keine Wirkung haben.

Sie können den Standardteams auch einige zusätzliche Berechtigungen Ihrer Wahl erteilen. So können Sie zum Beispiel allen „Benutzern“ die Berechtigung zum Verwalten von Bildschirmfotos erteilen.

Sie können auch einige neue benutzerdefinierte Teams definieren. Wenn Sie Ihre Berechtigungen für diese Teams weiterhin plattformweit verwalten möchten, wählen Sie einen geeigneten Wert für *Projektauswahl* (z. B. *Alle Projekte* oder *Alle öffentlichen Projekte*).

Eigene Berechtigungen für Sprachen, Komponenten oder Projekte

Sie können Ihre eigenen Teams erstellen, um die Berechtigungen für bestimmte Objekte wie Sprachen, Komponenten und Projekte zu verwalten. Obwohl diese Teams nur zusätzliche Berechtigungen gewähren, können Sie keine Berechtigung, die von plattformweiten oder projektspezifischen Teams gewährt wurde, durch Hinzufügen einer anderen benutzerdefinierten Team aufheben.

Beispiel:

Wenn Sie (aus welchen Gründen auch immer) die Übersetzung in eine bestimmte Sprache (sagen wir mal *Tschechisch*) nur einer geschlossenen Gruppe von zuverlässigen Übersetzern erlauben wollen, während die Übersetzungen in andere Sprachen öffentlich bleiben, müssen Sie das tun:

1. Entziehen Sie allen Benutzern die Erlaubnis, *Tschechisch* zu übersetzen. In der Standardkonfiguration kann dies durch Ändern des *Benutzer-Standardteams* erfolgen.

Tab. 1: Gruppe *Benutzer*

Sprachauswahl	Wie vorgegeben
Sprachen	Alle außer <i>Tschechisch</i>

2. Fügen Sie eine eigenes Team für *Tschechisch*-Übersetzer hinzu.

Tab. 2: Gruppe *Tschechische Übersetzer*

Rollen	<i>Hauptbenutzer</i>
Projektauswahl	<i>Alle öffentlichen Projekte</i>
Sprachauswahl	Wie vorgegeben
Sprachen	<i>Tschechisch</i>

3. Fügen Sie die Benutzer, denen Sie die Berechtigungen erteilen möchten, zu diesem Team hinzu.

Wie Sie sehen, ist die Verwaltung von Berechtigungen auf diese Weise zwar sehr leistungsfähig, kann aber auch ziemlich mühsam sein. Sie können sie nicht an einen anderen Benutzer delegieren, es sei denn, Sie erteilen Superuser-Berechtigungen.

Benutzer, Rollen, Teams, und Berechtigungen

Die Authentifizierungsmodelle bestehen aus verschiedenen Objekten:

Berechtigung

Durch Weblate bestimmte individuelle Berechtigung. Berechtigungen können nicht an Benutzer vergeben werden. Dies kann nur über die Zuweisung von Rollen erfolgen.

Rolle

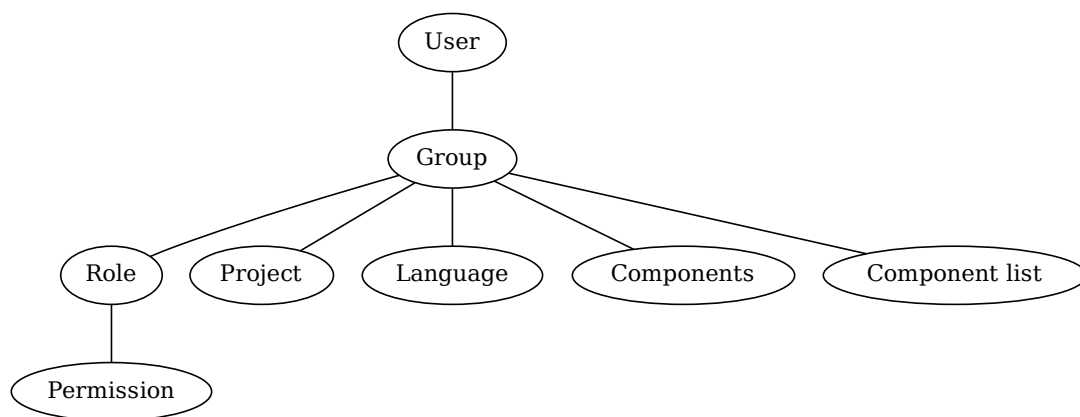
Eine Rolle bestimmt eine Zusammenstellung von Genehmigungen. Dies ermöglicht die Wiederverwendung dieser Zusammenstellungen an verschiedenen Orten, zur Vereinfachung der Verwaltung.

Benutzer

Der Benutzer kann zu mehreren Teams gehören.

Gruppe

Die Gruppe verbindet Rollen, Benutzer und Authentifizierungsobjekte (Projekte, Sprachen und Komponentenlisten).



Bemerkung: Einem Team können auch keine Rollen zugewiesen werden. In diesem Fall wird davon ausgegangen, dass jeder das Projekt durchsuchen kann (siehe unten).

Kontrolle des Zugriffs auf ein Projekt

Ein Benutzer muss Mitglied eines Teams sein, die mit dem Projekt oder einer Komponente in diesem Projekt verknüpft ist. Eine Mitgliedschaft allein genügt, es sind keine speziellen Genehmigungen erforderlich, um ein Projekt zu durchsuchen (dies wird in dem Standardteam *Betrachter* angewendet, siehe [Liste der Teams](#)).

Kontrolle des Zugriffs auf eine Komponente

Ein Benutzer kann auf nicht eingeschränkte Komponenten zugreifen, sobald er auf das Projekt der Komponente zugreifen kann (und über alle Rechte verfügt, die dem Benutzer für das Projekt gewährt wurden). Wenn *Eingeschränkter Zugriff* eingeschaltet ist, erfordert der Zugriff auf die Komponente explizite Berechtigungen für die Komponente (oder eine Komponentenliste, in der die Komponente enthalten ist).

Umfang der Teams

Der Umfang der von den Rollen in den Teams zugewiesenen Berechtigungen wird nach den folgenden Regeln angewendet:

- Wenn das Team eine *Komponentenliste* angibt, werden alle Berechtigungen, die den Mitgliedern dieses Teams gewährt werden, für alle Komponenten in den Komponentenlisten, die dem Team zugeordnet sind, gewährt, und ein Zugriff ohne zusätzliche Berechtigungen wird für alle Projekte gewährt, in denen sich diese Komponenten befinden. *Komponenten* und *Projekte* werden ignoriert.
- Wenn das Team ein *Komponenten* angibt, werden alle Berechtigungen, die den Mitgliedern dieses Teams erteilt wurden, für alle Komponenten gewährt, die dem Team zugeordnet sind, und ein Zugriff ohne zusätzliche Berechtigungen wird für alle Projekte gewährt, in denen diese Komponenten enthalten sind. *Projekte* werden ignoriert.
- Andernfalls, wenn das Team irgendwelche *Projekte* angibt, entweder durch direkte Auflistung oder indem *Projektauswahl* auf einen Wert wie *Alle öffentlichen Projekte* gesetzt wird, werden alle diese Berechtigungen auf alle Projekte angewandt, was effektiv dieselben Zugriffsrechte für alle *nicht eingeschränkten Komponenten* des Projekts gewährt.
- Die durch das *Sprachen* eines Teams auferlegten Einschränkungen, werden separat angewandt, wenn überprüft wird, ob ein Benutzer die Berechtigung für bestimmte durchzuführende Aktionen hat. Sie werden nämlich nur auf Aktionen angewendet, die direkt mit dem Übersetzungsprozess selbst zusammenhängen, wie zum Beispiel das Überprüfen, Speichern von Übersetzungen, Hinzufügen von Vorschlägen usw.

Hinweis: Verwenden Sie *Sprachauswahl* oder *Projektauswahl* um automatisch alle Sprachen oder Projekte einzuschließen.

Beispiel:

Angenommen, es gibt ein Projekt `foo` mit den Komponenten: `foo/bar` und `foo/baz` und das folgende Team:

Tab. 3: Gruppe `Administrator-Prüfer Spanisch`

Rollen	<i>Zeichenketten überprüfen, Repository verwalten</i>
Komponenten	<code>foo/bar</code>
Sprachen	<i>Spanisch</i>

Die Mitglieder dieses Teams haben folgende Rechte (unter Annahme der Standardrolleneinstellungen):

- Allgemeiner Zugriff (zur Durchsicht) auf das gesamte Projekt `foo` einschließlich der beiden Komponenten darin: `foo/bar` und `foo/baz`.
- Zeichenketten in der spanischen Übersetzung von `foo/bar` überprüfen (nirgendwo sonst).
- VCS für das gesamte `foo/bar`-Repository verwalten, z. B. ausstehende Änderungen von Übersetzern für alle Sprachen committen.

Automatische Teamzuweisungen

Unten auf der Bearbeitungsseite *Gruppe* in der *Django-Administrationsoberfläche* können Sie *Automatische Teamzuweisungen* angeben, das ist eine Liste regulärer Ausdrücke, die verwendet wird, um neu erstellte Benutzer anhand ihrer E-Mail-Adressen automatisch einem Team zuzuordnen. Diese Zuweisung erfolgt nur bei der Benutzerkontoerstellung.

Der häufigste Anwendungsfall für diese Funktion besteht darin, alle neuen Benutzer einem Standardteam zuzuordnen. Um dies zu erreichen, werden Sie wahrscheinlich den Standardwert (`^.*$`) im Feld für den regulären Ausdruck beibehalten wollen. Ein weiterer Anwendungsfall für diese Option könnte darin bestehen, den Mitarbeitern Ihres Unternehmens standardmäßig einige zusätzliche Privilegien zu gewähren. Unter der Annahme, dass alle Mitarbeiter eine Firmen-E-Mail-Adresse in Ihrer Domäne verwenden, lässt sich dies mit einem Ausdruck wie `^.*@mycompany.com` bewerkstelligen.

Bemerkung: Die automatische Teamzuweisung zu ‚Benutzer‘ und ‚Betrachter‘ wird immer neu erstellt, wenn Sie von einer Weblate-Version auf eine andere aktualisieren. Wenn Sie dies abstellen möchten, setzen Sie den regulären Ausdruck auf `^$` (der auf nichts zutrifft).

Bemerkung: Im Moment gibt es noch keine Möglichkeit, bereits bestehende Benutzer über die Benutzeroberfläche zu einem Team hinzuzufügen. Hierfür können Sie auf die [REST API](#) zurückgreifen.

Standardteams und -rollen

Nach der Installation wird ein Standardsatz von Teams erstellt (Siehe [Liste der Teams](#)).

Diese Rollen und Teams werden bei der Installation erstellt. Die eingebauten Rollen werden durch die Datenbankmigration beim Upgrade immer auf dem neuesten Stand gehalten. Sie können sie nicht ändern. Bitte definieren Sie eine neue Rolle, wenn Sie Ihre eigenen Berechtigungen festlegen möchten.

Liste der Berechtigungen und integrierten Rollen

Zielgruppe	Berechtigung	Rollen
Abrechnung (siehe Abrechnung)	Abrechnungsinformationen anzeigen	Verwaltung, Abrechnung
Änderungen	Änderungen herunterladen	Verwaltung
Kommentare	Kommentar erstellen	Verwaltung, Quelle bearbeiten, F
	Kommentar löschen	Verwaltung
	Kommentar lösen	Verwaltung, Zeichenketten überp
	Komponenteneinstellungen bearbeiten	Verwaltung
Komponente	Komponente sperren, um Übersetzungen zu verhindern	Verwaltung
	Glossareintrag hinzufügen	Verwaltung, Glossar verwalten,
Glossar	Glossareintrag bearbeiten	Verwaltung, Glossar verwalten,
	Glossareintrag löschen	Verwaltung, Glossar verwalten,
	Glossareinträge hochladen	Verwaltung, Glossar verwalten,
	Automatische Vorschläge verwenden	Verwaltung, Quelle bearbeiten, F
Übersetzungsspeicher	Übersetzungsspeicher bearbeiten	Verwaltung, Übersetzungsspeiche
	Übersetzungsspeicher löschen	Verwaltung, Übersetzungsspeiche
Projekte	Projekteinstellungen bearbeiten	Verwaltung
	Projektzugriff verwalten	Verwaltung
Auswertungen	Berichte herunterladen	Verwaltung
Bildschirmfotos	Bildschirmfoto hinzufügen	Verwaltung, Bildschirmfotos ver
	Bildschirmfoto bearbeiten	Verwaltung, Bildschirmfotos ver
	Bildschirmfoto löschen	Verwaltung, Bildschirmfotos ver

Tab. 4 – Fortsetzung der vorherigen

Zielgruppe	Berechtigung	Rollen
Ausgangszeichenketten	Zusätzliche Zeichenketteninformationen bearbeiten	Verwaltung, Quelle bearbeiten
Zeichenketten	Neue Zeichenkette hinzufügen	Verwaltung
	Entfernen einer Zeichenkette	Verwaltung
	Fehlerhafte Prüfung ablehnen	Verwaltung, Quelle bearbeiten, F
	Zeichenketten bearbeiten	Verwaltung, Quelle bearbeiten, F
	Zeichenketten überprüfen	Verwaltung, Zeichenketten überp
	Zeichenkette bearbeiten, wenn Vorschläge erzwungen werden	Verwaltung, Zeichenketten überp
	Ausgangszeichenketten bearbeiten	Verwaltung, Quelle bearbeiten, F
Vorschläge	Vorschlag annehmen	Verwaltung, Quelle bearbeiten, F
	Vorschlag hinzufügen	Verwaltung, Quelle bearbeiten, V
	Vorschlag löschen	Verwaltung, Hauptbenutzer
	Abstimmung über Vorschlag	Verwaltung, Quelle bearbeiten, F
Übersetzungen	Sprache zur Übersetzung hinzufügen	Verwaltung, Hauptbenutzer, Spra
	Automatische Übersetzung durchführen	Verwaltung, Automatische Vorsc
	Vorhandene Übersetzung löschen	Verwaltung, Sprachen verwalten
	Übersetzungsdatei herunterladen	Verwaltung, Quelle bearbeiten, A
	Mehrere Sprachen zur Übersetzung hinzufügen	Verwaltung, Sprachen verwalten
Uploads	Autor der hochgeladenen Übersetzung definieren	Verwaltung
	Vorhandene Zeichenketten mit Hochladung überschreiben	Verwaltung, Quelle bearbeiten, F
VCS	Übersetzungen hochladen	Verwaltung, Quelle bearbeiten, F
	Zugriff auf das interne Repository	Verwaltung, Auf Repository zugr
	Commit für Änderungen am internen Repository durchführen	Verwaltung, Repository verwalte
	Änderung aus dem internen Repository pushen	Verwaltung, Repository verwalte
	Änderungen im internen Repository zurücksetzen	Verwaltung, Repository verwalte
	Speicherort des Upstream-Repositorys anzeigen	Verwaltung, Auf Repository zugr
	Aktualisierung des internen Repositorys	Verwaltung, Repository verwalte
Plattformweite Berechtigungen	Verwaltungsoberfläche verwenden	
	Neue Projekte hinzufügen	
	Sprachdefinitionen hinzufügen	
	Sprachdefinitionen verwalten	
	Teams verwalten	
	Benutzer verwalten	
	Rollen verwalten	
	Mitteilungen verwalten	
	Übersetzungsspeicher verwalten	
	Übersetzungsmaschinen verwalten	
	Komponentenlisten verwalten	

Bemerkung: Plattformweite Berechtigungen werden keiner Standardrolle gewährt. Diese sind sehr mächtig und kommen dem Superuser-Status recht nahe. Die meisten von ihnen betreffen alle Projekte in Ihrer Weblate-Installation.

Liste der Teams

Die folgenden Teams werden bei der Installation (oder nach dem Ausführen von `setupgroups`) erstellt, und Sie können diese nach Belieben ändern. Wenn Sie die Gruppen jedoch löschen oder umbenennen, werden sie bei der Migration neu erstellt.

Gäste

Legt die Berechtigungen für nicht authentifizierte Benutzer fest.

Dieses Team enthält ausschließlich anonyme Benutzer (siehe `ANONYMOUS_USER_NAME`).

Sie können Rollen aus diesem Team entfernen, um die Berechtigungen für nicht-authentifizierte Benutzer einzuschränken.

Standard-Rollen: *Vorschlag hinzufügen, Auf Repository zugreifen*

Betrachter

Diese Rolle stellt die Sichtbarkeit der öffentlichen Projekte für alle Benutzer sicher. Standardmäßig sind alle Benutzer Mitglieder dieses Teams.

Standardmäßig werden alle neuen Konten durch *Automatische Teamzuweisungen* zu Mitgliedern dieses Teams, wenn sie ihr beitreten.

Standard-Rollen: keine

Benutzer

Standardteam für alle Benutzer.

Standardmäßig werden alle neuen Konten durch *Automatische Teamzuweisungen* zu Mitgliedern dieses Teams, wenn sie ihr beitreten.

Standard-Rollen: *Hauptbenutzer*

Prüfer

Gruppe für Prüfer (siehe *Übersetzungsabläufe*).

Standard-Rollen: *Zeichenketten überprüfen*

Manager

Gruppe für Administratoren.

Standard-Rollen: *Verwaltung*

Warnung: Entfernen Sie niemals die vordefinierten Weblate-Teams und -Benutzer, da dies zu unerwarteten Problemen führen kann! Wenn Sie keine Verwendung für sie haben, können Sie stattdessen alle ihre Berechtigungen entfernen.

2.6.3 Zusätzliche Zugriffsbeschränkungen

Wenn Sie Ihre Weblate-Installation weniger öffentlich nutzen wollen, d. h. neue Benutzer nur per Einladung zulassen möchten, können Sie Weblate so konfigurieren, dass nur bekannte Benutzer Zugriff darauf haben. Dazu müssen Sie `REGISTRATION_OPEN` auf `False` setzen, um die Registrierung neuer Benutzer zu verhindern, und `REQUIRE_LOGIN` auf `/*` setzen, um eine Anmeldung für den Zugriff auf alle Seiten der Website zu verlangen. Dies ist im Grunde der Weg, um Ihre Weblate-Installation zu sperren.

Hinweis: Um neue Benutzer hinzuzufügen können Sie die integrierte Funktion *Neue Benutzereinladung* verwenden.

2.7 Übersetzungsprojekte

2.7.1 Translation organization

Weblate organizes translatable VCS content of project/components into a tree-like structure.

- The bottom level object is *Projektkonfiguration*, which should hold all translations belonging together (for example translation of an application in several versions and/or accompanying documentation).
- On the level above, *Komponentenkonfiguration*, which is actually the component to translate, you define the VCS repository to use, and the mask of files to translate.
- Above *Komponentenkonfiguration* there are individual translations, handled automatically by Weblate as translation files (which match *Dateimaske* defined in *Komponentenkonfiguration*) appear in the VCS repository.

Weblate supports a wide range of translation formats (both bilingual and monolingual ones) supported by Translate Toolkit, see [Unterstützte Dateiformate](#).

Bemerkung: You can share cloned VCS repositories using [Weblate-interne URLs](#). Using this feature is highly recommended when you have many components sharing the same VCS. It improves performance and decreases required disk space.

2.7.2 Adding translation projects and components

Geändert in Version 3.2: An interface for adding projects and components is included, and you no longer have to use [Die Django-Adminoberfläche](#).

Geändert in Version 3.4: The process of adding components is now multi staged, with automated discovery of most parameters.

Based on your permissions, new translation projects and components can be created. It is always permitted for users with the *Add new projects* permission, and if your instance uses billing (e.g. like <https://hosted.weblate.org/> see [Abrechnung](#)), you can also create those based on your plans allowance from the user account that manages billing.

You can view your current billing plan on a separate page:

Navigation: Weblate Dashboard Projects Languages Checks

Profile: Your profile / Billing

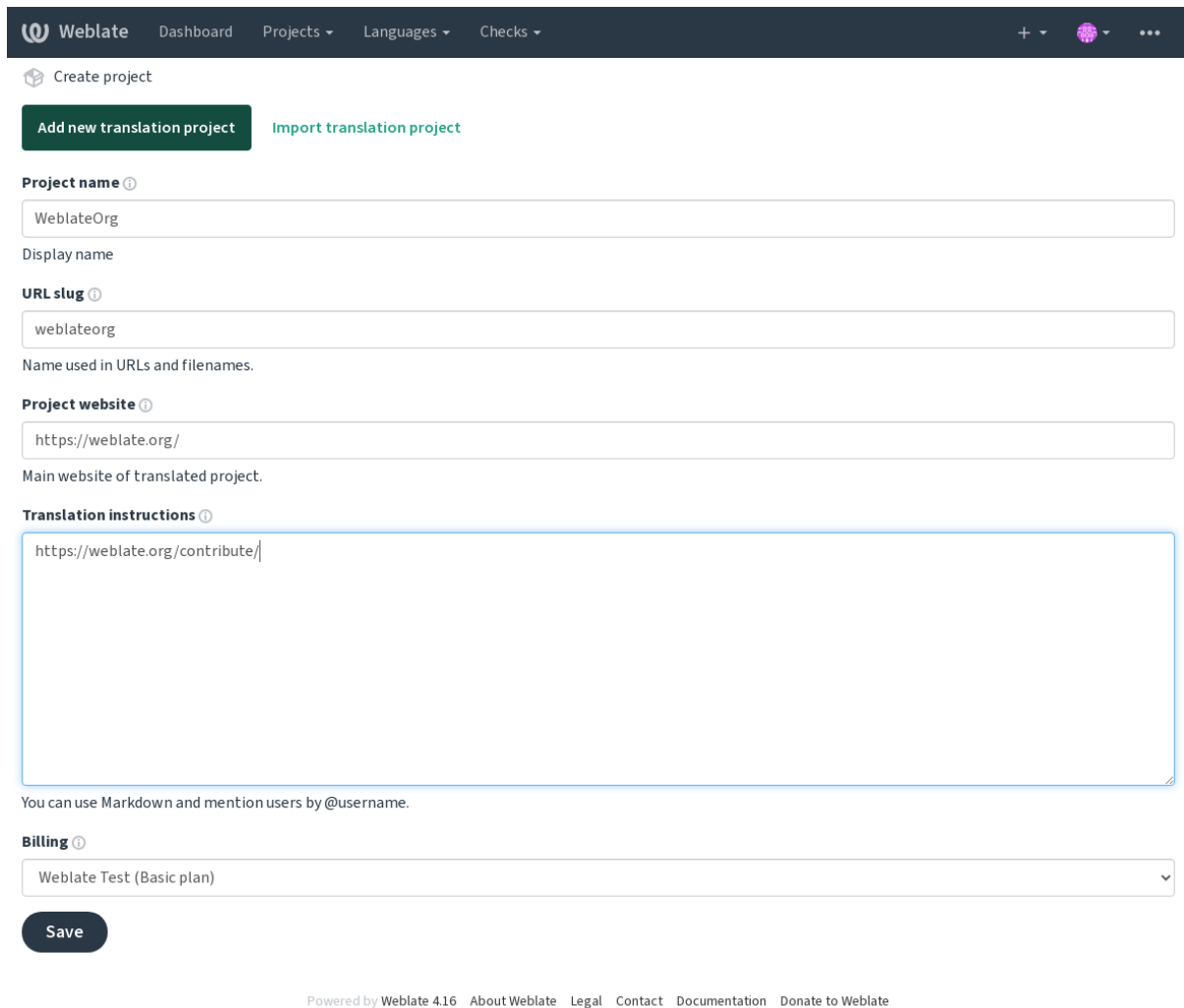
Billing plan	
Current plan	Basic plan (Active)
Monthly price	19 EUR
Yearly price	199 EUR
Strings limit	Used 0
Languages limit	Used 0
Last invoice	2023-02-28 - 2023-03-02
Projects limit	Used 0 of 1
Projects	No projects currently assigned!

Buttons: Add new translation project, Terminate billing plan

Invoices		
Invoice period	Invoice amount	Download invoice
02/28/2023 - 03/02/2023	19.0 EUR	Not available

Footer: Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

The project creation can be initiated from there, or using the menu in the navigation bar, filling in basic info about the translation project to complete addition of it:



Create project

Add new translation project **Import translation project**

Project name ⓘ
WeblateOrg

Display name

URL slug ⓘ
weblateorg

Name used in URLs and filenames.

Project website ⓘ
https://weblate.org/

Main website of translated project.

Translation instructions ⓘ
https://weblate.org/contribute/

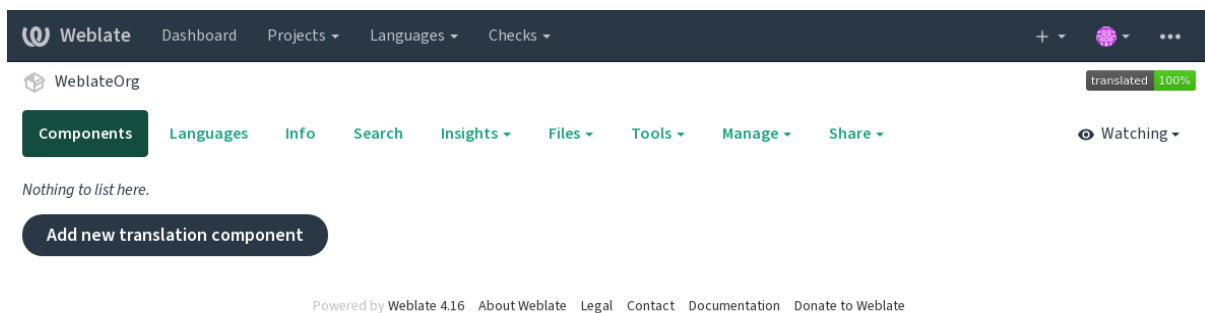
You can use Markdown and mention users by @username.

Billing ⓘ
Weblate Test (Basic plan)

Save

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

After creating the project, you are taken directly to the project page:



WeblateOrg translated 100%

Components Languages Info Search Insights Files Tools Manage Share

Watching

Nothing to list here.

Add new translation component

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Creating a new translation component can be initiated via a single click there. The process of creating a component is multi-staged and automatically detects most translation parameters. There are several approaches to creating component:

Von der Versionskontrolle

Creates component from remote version control repository.

Aus vorhandener Komponente

Creates additional component to existing one by choosing different files.

Zusätzlicher Zweig

Creates additional component to existing one, just for different branch.

Übersetzungsdateien hochladen

Upload translation files to Weblate in case you do not have version control or do not want to integrate it with Weblate. You can later update the content using the web interface or [Weblates REST API](#).

Dokument übersetzen

Laden Sie ein einzelnes Dokument oder eine Übersetzungsdatei hoch und übersetzen Sie diese.

Bei Null anfangen

Create blank translation project and add strings manually.

Once you have existing translation components, you can also easily add new ones for additional files or branches using same repository.

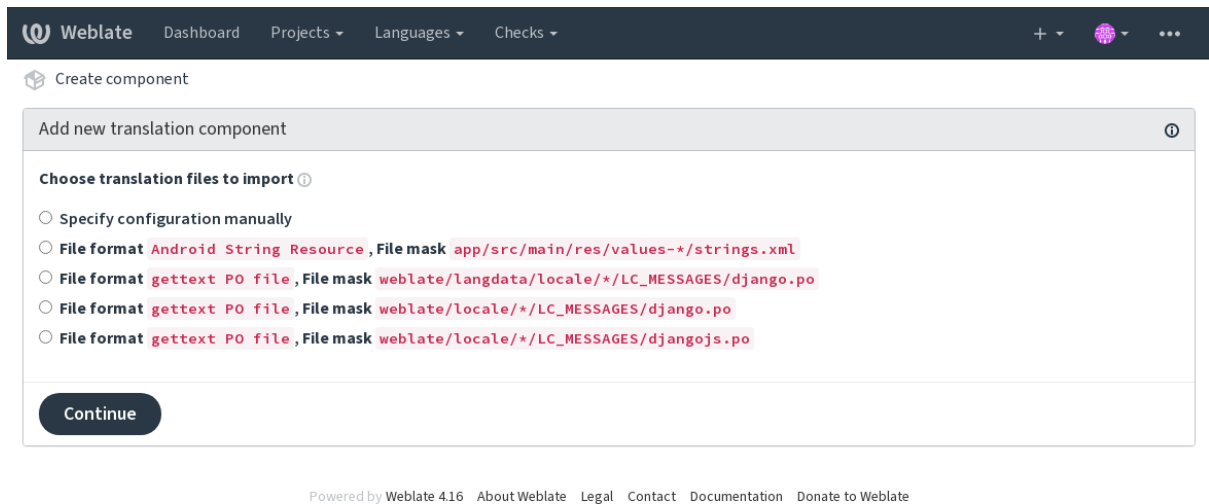
First you need to fill in name and repository location:

The screenshot shows the 'Create component' form in the Weblate web interface. The form is titled 'Create component' and has a dark header bar with the Weblate logo and navigation links: 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below the header, there are four tabs: 'From version control' (selected), 'Upload translations files', 'Translate document', and 'Start from scratch'. The form fields are as follows:

- Component name**: A text input field containing 'Language names'.
- Display name**: A text input field.
- URL slug**: A text input field containing 'language-names'.
- Name used in URLs and filenames.**: A text input field.
- Use as a glossary**: A checkbox that is unchecked.
- Project**: A dropdown menu showing 'WeblateOrg'.
- Source language**: A dropdown menu showing 'English'.
- Language used for source strings in all components**: A text input field.
- Version control system**: A dropdown menu showing 'Git'.
- Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.**: A text input field.
- Source code repository**: A text input field containing 'https://github.com/WeblateOrg/demo.git'.
- URL of a repository, use weblate://project/component to share it with other component.**: A text input field.
- Repository branch**: A text input field.
- Repository branch to translate**: A text input field.

At the bottom of the form, there is a 'Continue' button. Below the form, there is a footer bar with the text 'Powered by Weblate 4.16' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

On the next page, you are presented with a list of discovered translatable resources:

 Weblate Dashboard Projects Languages Checks

Create component

Add new translation component

Choose translation files to import

☐ Specify configuration manually

☐ File format `Android String Resource`, File mask `app/src/main/res/values-*/strings.xml`

☐ File format `gettext PO file`, File mask `weblate/langdata/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/djangojs.po`

Continue

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

As a last step, you review the translation component info and fill in optional details:

Web

late

Dashboard

Projects

Languages

Checks

Create component

Detected license as MIT, please check whether it is correct.

Add new translation component

Project

Web

lateOrg

Component name

Language names

Display name

URL slug

language-names

Name used in URLs and filenames.

Version control system

Git

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository

https://github.com/Web

lateOrg/demo.git

URL of a repository, use weblate://project/component to share it with other component.

Repository branch

Repository branch to translate

Repository push URL

URL of a push repository, pushing is turned off if empty.

Push branch

Branch for pushing changes, leave empty to use repository branch

Repository browser

https://github.com/Web

lateOrg/demo/blob/{{branch}}/{{filename}}#L{{line}}

Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. You might want to strip leading directory by using {{filename|parentdir}}.

File format

gettext PO file

File mask

app/src/main/res/values-*/strings.xmlweblate/langdata/locale/*/LC_MESSAGES/django.po

Path of files to translate relative to repository root, use * instead of language code, for example: po/* .po or locale/*/LC_MESSAGES/django.po.

Monolingual base language file

app/src/main/res/values/strings.xml

Filename of translation base file, containing all strings and their source; it is recommended for monolingual translation formats.

☒ Edit base file

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Adding new translation

Create new language file

How to handle requests for creating new translations.

Template for new translations

weblate/langdata/locale/django.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Translation license

GNU General Public License v3.0 or later

Language code style

Default based on the file format

Customize language code used to generate the filename for translations created by Weblate.

Language filter

^(cs|he|hu)\$

Regular expression used to filter translation files when scanning for file mask.

Source language

English

Language used for source strings in all components

☐ Use as a glossary

You will be able to edit more options in the component settings after creating it.

Save

Siehe auch:

Die Django-Adminoberfläche, Projektkonfiguration, Komponentenkonfiguration

2.7.3 Projektkonfiguration

Create a translation project and then add a new component for translation in it. The project is like a shelf, in which real translations are stacked. All components in the same project share suggestions and their dictionary; the translations are also automatically propagated through all components in a single project (unless turned off in the component configuration), see *Übersetzungsspeicher*.

Siehe auch:

/devel/integration

These basic attributes set up and inform translators of a project:

Projektname

Verbose project name, used to display the project name.

Siehe auch:

PROJECT_NAME_RESTRICT_RE

URL-Kurzbegriff

Project name suitable for URLs.

Projektseite

URL where translators can find more info about the project.

This is a required parameter unless turned off by *WEBSITE_REQUIRED*.

Siehe auch:

PROJECT_WEB_RESTRICT_RE

Übersetzungsanweisungen

Text describing localization process in the project, and any other information useful for translators. Markdown can be used for text formatting or inserting links.

Kopfzeile „Language-Team“ setzen

Ob Weblate den Language-Team Header verwalten soll (dies ist zur Zeit nur eine *GNU gettext* Funktion).

Gemeinsamen Übersetzungsspeicher verwenden

Whether to use shared translation memory, see *Geteilter Übersetzungsspeicher* for more details.

Der Standardwert kann durch `DEFAULT_SHARED_TM` geändert werden.

Zu einem gemeinsamen Übersetzungsspeicher beitragen

Whether to contribute to shared translation memory, see *Geteilter Übersetzungsspeicher* for more details.

Der Standardwert kann durch `DEFAULT_SHARED_TM` geändert werden.

Zugriffssteuerung

Configure per project access control, see *Projekt-Zugriffssteuerung* for more details.

Der Standardwert kann durch `DEFAULT_ACCESS_CONTROL` geändert werden.

Überprüfung aktivieren

Enable review workflow for translations, see *Zugehörige Prüfer*.

Quellenüberprüfung aktivieren

Enable review workflow for source strings, see *Überprüfungen der Ausgangszeichenketten*.

Siehe auch:

report-source, *Kommentare*

Hooks aktivieren

Whether unauthenticated *Benachrichtigungs-Hooks* are to be used for this repository.

Siehe auch:

Zwischensprachdatei, Qualitäts-Gateway für die Ausgangszeichenketten, Zweisprachige und einsprachige Formate, Sprachdefinitionen

Sprachaliasnamen

Define language codes mapping when importing translations into Weblate. Use this when language codes are inconsistent in your repositories and you want to get a consistent view in Weblate or in case you want to use non-standard naming of your translation files.

The typical use case might be mapping American English to English: `en_US:en`

Multiple mappings to be separated by comma: `en_GB:en, en_US:en`

Using non standard code: `ia_FOO:ia`

Hinweis: The language codes are mapped when matching the translation files and the matches are case sensitive, so make sure you use the source language codes in same form as used in the filenames.

Siehe auch:

Neue Übersetzungen hinzufügen, Sprachcode, Parsing von Sprachcodes

2.7.4 Komponentenkonfiguration

A component is a grouping of something for translation. You enter a VCS repository location and file mask for which files you want translated, and Weblate automatically fetches from this VCS, and finds all matching translatable files.

Siehe auch:

/devel/integration

You can find some examples of typical configurations in the *Unterstützte Dateiformate*.

Bemerkung: It is recommended to keep translation components to a reasonable size - split the translation by anything that makes sense in your case (individual apps or add-ons, book chapters or websites).

Weblate easily handles translations with 10000s of strings, but it is harder to split work and coordinate among translators with such large translation components.

Should the language definition for a translation be missing, an empty definition is created and named as „cs_CZ (generated)“. You should adjust the definition and report this back to the Weblate authors, so that the missing languages can be included in next release.

The component contains all important parameters for working with the VCS, and for getting translations out of it:

Name der Komponente

Verbose component name, used to display the component name.

Komponenten-Slug

Component name suitable for URLs.

Komponenten-Projekt

Projekt, zu dem die Komponente gehört.

Versionsverwaltung

VCS to use, see *Integration der Versionsverwaltung* for details.

Siehe auch:

Pushen von Änderungen aus Weblate

Quellcode-Repository

VCS repository used to pull changes.

Siehe auch:

See *Zugriff auf Repositories* for more details on specifying URLs.

Hinweis: This can either be a real VCS URL or `weblate://project/component` indicating that the repository should be shared with another component. See *Weblate-interne URLs* for more details.

Push-URL für Repository

Repository URL used for pushing. This setting is used only for *Git* and *Mercurial* and push support is turned off for these when this is empty.

For linked repositories, this is not used and setting from linked component applies.

Siehe auch:

See *Zugriff auf Repositories* for more details on how to specify a repository URL and *Pushen von Änderungen aus Weblate* for more details on pushing changes from Weblate.

Paketquellnavigator

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use *Template markup*.

For example on GitHub, use something like: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

Falls Ihre Pfade relativ zu verschiedenen Verzeichnissen sind (Pfad enthält `..`), sollten Sie das führende Verzeichnis durch den Filter `parentdir` entfernen (siehe *Template markup*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

Exportierte Paketquelladresse

URL where changes made by Weblate are exported. This is important when *Kontinuierliche Lokalisierung* is not used, or when there is a need to manually merge changes. You can use *Git-Exporter* to automate this for Git repositories.

Repository-Zweig

Which branch to checkout from the VCS, and where to look for translations.

For linked repositories, this is not used and setting from linked component applies.

Push Branch

Branch for pushing changes, leave empty to use *Repository-Zweig*.

For linked repositories, this is not used and setting from linked component applies.

Bemerkung: This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

Siehe auch:

Pushen von Änderungen aus Weblate

Dateimaske

Mask of files to translate, including path. It should include one „*“ replacing language code (see *Sprachdefinitionen* for info on how this is processed). In case your repository contains more than one translation file (e.g. more gettext domains), you need to create a component for each of them.

Zum Beispiel `po/* .po` oder `locale/*/LC_MESSAGES/django.po`.

In case your filename contains special characters such as `[`, `]`, these need to be escaped as `[]` or `[]`.

Siehe auch:

Zweisprachige und einsprachige Formate, Was bedeutet „Es gibt mehrere Dateien für die einzelne Sprache (en)“?

Einsprachige Basis-Sprachdatei

Base file containing string definitions for *Einsprachige Komponenten*.

Siehe auch:

Zweisprachige und einsprachige Formate, Was bedeutet „Es gibt mehrere Dateien für die einzelne Sprache (en)“?

Basisdatei bearbeiten

Whether to allow editing the base file for *Einsprachige Komponenten*.

Zwischensprachdatei

Intermediate language file for *Einsprachige Komponenten*. In most cases this is a translation file provided by developers and is used when creating actual source strings.

When set, the source strings are based on this file, but all other languages are based on *Einsprachige Basis-Sprachdatei*. In case the string is not translated into the source language, translating to other languages is prohibited. This provides *Qualitäts-Gateway für die Ausgangszeichenketten*.

Siehe auch:

Qualitäts-Gateway für die Ausgangszeichenketten, Zweisprachige und einsprachige Formate, Was bedeutet „Es gibt mehrere Dateien für die einzelne Sprache (en)“?

Vorlage für neue Übersetzungen

Base file used to generate new translations, e.g. `.pot` file with gettext.

Hinweis: In many monolingual formats Weblate starts with empty file by default. Use this in case you want to have all strings present with empty value when creating new translation.

Siehe auch:

adding-translation, *Neue Übersetzungen hinzufügen, Neue Übersetzung hinzufügen, Zweisprachige und einsprachige Formate, Was bedeutet „Es gibt mehrere Dateien für die einzelne Sprache (en)“?*

Dateiformat

Translation file format, see also *Unterstützte Dateiformate*.

Adresse für Fehlerberichte bei Ausgangszeichenketten

Email address used for reporting upstream bugs. This address will also receive notification about any source string comments made in Weblate.

Verbreitung von Übersetzungen erlauben

You can turn off propagation of translations to this component from other components within same project. This really depends on what you are translating, sometimes it's desirable to have make use of a translation more than once.

It's usually a good idea to turn this off for monolingual translations, unless you are using the same IDs across the whole project.

Default value can be changed by `DEFAULT_TRANSLATION_PROPAGATION`.

Siehe auch:

Übersetzungen aller Komponenten gleich halten

Vorschläge aktivieren

Whether translation suggestions are accepted for this component.

Abstimmen über Vorschläge

Turns on vote casting for suggestions, see *Abstimmen über Vorschläge*.

Vorschläge automatisch annehmen

Automatically accept voted suggestions, see *Abstimmen über Vorschläge*.

Übersetzungsmarkierungen

Customization of quality checks and other Weblate behavior, see *Anpassen des Verhaltens mit Markierungen*.

Erzwungene Qualitätsprüfungen

Liste der Prüfungen, die nicht ignoriert werden können, siehe *Qualitätsprüfungen erzwingen*.

Bemerkung: Das Erzwingen der Prüfung aktiviert sie nicht automatisch, Sie sollten sie immer noch mit *Anpassen des Verhaltens mit Markierungen* in *Übersetzungsmarkierungen* oder *Zusätzliche Informationen über Ausgangszeichenketten* aktivieren.

Lizenzierung der Übersetzung

License of the translation (does not need to be the same as the source code license).

Mitwirkungsvereinbarung

Die Benutzervereinbarung muss genehmigt werden, bevor der Benutzer die Komponenten übersetzen kann.

Neue Übersetzung hinzufügen

How to handle requests for creation of new languages. Available options:

Maintainer kontaktieren

Der Benutzer kann die gewünschte Sprache auswählen und die Projektbetreuer erhalten eine Benachrichtigung darüber. Es liegt an ihnen, ob Sie die Sprache zum Repository hinzufügen.

Geben Sie die URL für die Übersetzungsanweisungen an

Der Benutzer erhält einen Link zu einer Seite, die den Prozess des Beginns neuer Übersetzungen beschreibt. Verwenden Sie diese Seite, wenn ein formellerer Prozess gewünscht wird (z. B. die Bildung eines Teams von Personen vor dem Beginn der eigentlichen Übersetzung).

Neue Sprachdatei erstellen

User can select language and Weblate automatically creates the file for it and translation can begin.

Hinzufügen neuer Übersetzungen deaktivieren

There will be no option for user to start new translation.

Hinweis: The project admins can add new translations even if it is disabled here when it is possible (either *Vorlage für neue Übersetzungen* or the file format supports starting from an empty file).

Siehe auch:

adding-translation, *Neue Übersetzungen hinzufügen*

Zeichenketten verwalten

Neu in Version 4.5.

Legt fest, ob Benutzer in Weblate neue Zeichenketten hinzufügen und bestehende entfernen dürfen. Passen Sie dies an Ihren Lokalisierungsprozess an – wie die neuen Zeichenketten eingeführt werden sollen.

For bilingual formats, the strings are typically extracted from the source code (for example by using `xgettext`) and adding new strings in Weblate should be disabled (they would be discarded next time you update the translation files). In Weblate you can manage strings for every translation and it does not enforce the strings in all translations to be consistent.

For monolingual formats, the strings are managed only on source language and are automatically added or removed in the translations. The strings appear in the translation files once they are translated.

Siehe auch:

Zweisprachige und einsprachige Formate, adding-new-strings, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Stil des Sprachcodes

Anpassen des Sprachcodes, der zur Generierung des Dateinamens für von Weblate erstellte Übersetzungen verwendet wird.

Siehe auch:

Neue Übersetzungen hinzufügen, Sprachcode, Parsing von Sprachcodes

Git-Strategie

You can configure how updates from the upstream repository are handled. The actual implementation depends on VCS, see *Integration der Versionsverwaltung*.

Umbasieren (rebase)

Rebases Weblate commits on top of upstream repository on update. This provides clean history without extra merge commits.

Rebasing can cause you trouble in case of complicated merges, so carefully consider whether or not you want to enable them.

You might need to enable force pushing by choosing *Git Push erzwingen* as *Versionsverwaltung*, especially when pushing to a different branch.

Zusammenführen

Upstream repository changes are merged into Weblate one. This setting utilizes fast-forward when possible. This is the safest way, but might produce a lot of merge commits.

Zusammenführen ohne Fast-Forward

Upstream repository changes are merged into Weblate one with doing a merge commit every time (even when fast-forward would be possible). Every Weblate change will appear as a merge commit in Weblate repository.

Default value can be changed by `DEFAULT_MERGE_STYLE`.

Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten

Message used when committing a translation, see *Template markup*.

Die Standardwerte können durch das setzen von `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE`, `DEFAULT_MERGE_MESSAGE` und `DEFAULT_PULL_MESSAGE` geändert werden.

Bei Commit gleichzeitig Pushen

Whether committed changes should be automatically pushed to the upstream repository. When enabled, the push is initiated once Weblate commits changes to its underlying repository (see *Lazy Commits*). To actually enable pushing *Repository push URL* has to be configured as well.

Alter der Änderungen, das erreicht sein muss, bevor ein Commit erfolgt

Sets how old (in hours) changes have to be before they are committed by background task or the `commit_pending` management command. All changes in a component are committed once there is at least one change older than this period.

Default value can be changed by `COMMIT_PENDING_HOURS`.

Hinweis: There are other situations where pending changes might be committed, see *Lazy Commits*.

Sperre bei Fehler

Locks the component (and linked components, see [Weblate-interne URLs](#)) upon the first failed push or merge into its upstream repository, or pull from it. This avoids adding another conflicts, which would have to be resolved manually.

The component will be automatically unlocked once there are no repository errors left.

Ausgangssprache

Language used for source strings. Change this if you are translating from something else than English.

Hinweis: In case you are translating bilingual files from English, but want to be able to do fixes in the English translation as well, choose *English (Developer)* as a source language to avoid conflict between the name of the source language and the existing translation.

For monolingual translations, you can use intermediate translation in this case, see [Zwischensprachedatei](#).

Sprachen-Filter

Regulärer Ausdruck, der zum Filtern der Übersetzung beim Scannen nach Dateimaske verwendet wird. Er kann verwendet werden, um die Liste der von Weblate verwalteten Sprachen einzuschränken.

Bemerkung: You need to list language codes as they appear in the filename.

Some examples of filtering:

Filter description	Regulärer Ausdruck
Selected languages only	<code>^(cs de es)\$</code>
Exclude languages	<code>^(?! (it fr) \$) .+\$</code>
Filter two letter codes only	<code>^[.]+\$</code>
Exclude non language files	<code>^(?! (blank) \$) .+\$</code>
Include all files (default)	<code>^[^.] +\$</code>

Varianten regulärer Ausdruck

Regular expression used to determine the variants of a string, see variants.

Bemerkung: Die meisten Felder können von Projektverantwortlichen oder Administratoren in der Weblate-Benutzeroberfläche bearbeitet werden.

Siehe auch:

Unterstützt Weblate auch andere VCS als Git und Mercurial?, alerts

Priorität

Komponenten mit höherer Priorität werden Übersetzern zuerst angeboten.

Geändert in Version 4.15: This now also affects ordering of matched glossary terms.

Eingeschränkter Zugriff

By default the component is visible to anybody who has access to the project, even if the person can not perform any changes in the component. This makes it easier to keep translation consistency within the project.

Restricting access at a component, or component-list level takes over access permission to a component, regardless of project-level permissions. You will have to grant access to it explicitly. This can be done through granting access to a new user group and putting users in it, or using the default *custom* or *private* access control groups.

The default value can be changed in `DEFAULT_RESTRICTED_COMPONENT`.

Hinweis: This applies to project admins as well — please make sure you will not loose access to the component after toggling the status.

In Projekten teilen

You can choose additional projects where the component will be visible. Useful for shared libraries which you use in several projects.

Bemerkung: Sharing a component doesn't change its access control. It only makes it visible when browsing other projects. Users still need access to the actual component to browse or translate it.

Verwendung als Glossar

Neu in Version 4.5.

Ermöglicht die Verwendung dieser Komponente als Glossar. Sie können mit *Glossarfarbe* konfigurieren, wie es aufgelistet werden soll.

Das Glossar ist in allen durch *In Projekten teilen* definierten Projekten zugänglich.

It is recommended to enable *Zeichenketten verwalten* on glossaries in order to allow adding new words to them.

Siehe auch:

Glossar

Glossarfarbe

Display color for a glossary used when showing word matches.

2.7.5 Template markup

Weblate uses simple markup language in several places where text rendering is needed. It is based on [The Django template language](#), so it can be quite powerful.

Currently it is used in:

- Commit message formatting, see *Komponentenkonfiguration*
- **Mehrere Erweiterungen**
 - *Komponentenerkennung*
 - *Statistikgenerator*
 - *Ausführen von Skripten der Erweiterung*

There following variables are available in the component templates:

```

{{ language_code }}
    Sprachcode

{{ language_name }}
    Sprachenname

{{ component_name }}
    Name der Komponente

{{ component_slug }}
    Komponenten-Slug

{{ project_name }}
    Projektname

{{ project_slug }}
    Projekt-Slug

{{ url }}
    Übersetzungs URL

{{ filename }}
    Übersetzungsdateiname

{{ stats }}
    Translation stats, this has further attributes, examples below.

{{ stats.all }}
    Total strings count

{{ stats.fuzzy }}
    Count of strings needing review

{{ stats.fuzzy_percent }}
    Percent of strings needing review

{{ stats.translated }}
    Translated strings count

{{ stats.translated_percent }}
    Translated strings percent

{{ stats.allchecks }}
    Number of strings with failing checks

{{ stats.allchecks_percent }}
    Percent of strings with failing checks

{{ author }}
    Author of current commit, available only in the commit scope.
```

`{{ addon_name }}`

Name of currently executed add-on, available only in the add-on commit message.

The following variables are available in the repository browser or editor templates:

`{{branch}}`

Aktueller Zweig

`{{line}}`

line in file

`{{filename}}`

filename, you can also strip leading parts using the `parentdir` filter, for example `{{filename|parentdir}}`

You can combine them with filters:

```
{{ component|title }}
```

You can use conditions:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

There is additional tag available for replacing characters:

```
{% replace component "-" " " %}
```

You can combine it with filters:

```
{% replace component|capfirst "-" " " %}
```

There are also additional filter to manipulate with filenames:

```
Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|striptext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times: {{ filename|parentdir|parentdir }}
```

... und andere Django-Vorlagenfunktionen.

2.7.6 Importierungsgeschwindigkeit

Fetching VCS repository and importing translations to Weblate can be a lengthy process, depending on size of your translations. Here are some tips:

Optimize configuration

The default configuration is useful for testing and debugging Weblate, while for a production setup, you should do some adjustments. Many of them have quite a big impact on performance. Please check *Produktionseinrichtung* for more details, especially:

- Configure Celery for executing background tasks (see *Hintergrundaufgaben mit Celery*)
- *Caching einschalten*
- *Eine leistungsstarke Datenbank-Engine verwenden*
- *Debugmodus deaktivieren*

Check resource limits

If you are importing huge translations or repositories, you might be hit by resource limitations of your server.

- Check the amount of free memory, having translation files cached by the operating system will greatly improve performance.
- Disk operations might be bottleneck if there is a lot of strings to process—the disk is pushed by both Weblate and the database.
- Additional CPU cores might help improve performance of background tasks (see *Hintergrundaufgaben mit Celery*).

Disable unneeded checks

Some quality checks can be quite expensive, and if not needed, can save you some time during import if omitted. See *CHECK_LIST* for info on configuration.

2.7.7 Automatic creation of components

In case your project has dozen of translation files (e.g. for different gettext domains, or parts of Android apps), you might want to import them automatically. This can either be achieved from the command-line by using *import_project* or *import_json*, or by installing the *Komponentenerkennung* add-on.

To use the add-on, you first need to create a component for one translation file (choose the one that is the least likely to be renamed or removed in future), and install the add-on on this component.

For the management commands, you need to create a project which will contain all components and then run *import_project* or *import_json*.

Siehe auch:

Management commands, Komponentenerkennung

2.8 Sprachdefinitionen

Um verschiedene Übersetzungen korrekt darzustellen, werden Informationen über den Namen der Sprache, die Leserichtung, die Pluraldefinitionen und den Sprachcode benötigt.

2.8.1 Integrierte Sprachdefinitionen

Definitionen für etwa 600 Sprachen sind in Weblate enthalten und die Liste wird in jeder Version erweitert. Jedes Mal, wenn Weblate aktualisiert wird (genauer gesagt, wenn **weblate migrate** ausgeführt wird, siehe *Allgemeine Upgrade-Anweisungen*), wird die Datenbank der Sprachen aktualisiert, um alle in Weblate enthaltenen Sprachdefinitionen aufzunehmen.

Diese Funktion kann mit *UPDATE_LANGUAGES* abgeschaltet werden. Mit *setuplang* können Sie auch die Aktualisierung der Datenbank erzwingen, damit sie mit den in Weblate integrierten Daten übereinstimmt.

Siehe auch:

Erweiterung der integrierten Sprachdefinitionen, Aktuelle Sprachdefinitionen

2.8.2 Parsing von Sprachcodes

Während des Parsens von Übersetzungen versucht Weblate, den Sprachcode (in der Regel den ISO 639-1 Code) aus der *Dateimask*e auf ein vorhandenes Sprachobjekt abzubilden.

Sie können diese Zuordnung auf Projektebene durch *Sprachaliasnamen* weiter anpassen.

Wenn keine exakte Übereinstimmung gefunden werden kann, wird versucht, sie bestmöglich in eine bestehende Sprache einzupassen. Die folgenden Schritte werden versucht:

- Suchen ohne Berücksichtigung der Groß-/Kleinschreibung.
- Normalisierung von Unterstrichen und Bindestrichen.
- Suche nach integrierten Sprachaliasen.
- Nach Sprachennamen suchen.
- Ignorieren des Standard-Ländercodes für eine bestimmte Sprache – Auswahl von `cs` anstelle von `cs_CZ`.

Sollte auch dies fehlschlagen, wird eine neue Sprachdefinition unter Verwendung der Standardeinstellungen (Textrichtung von links nach rechts, ein Plural) erstellt. Die automatisch erstellte Sprache mit dem Code `xx_XX` wird als *xx_XX (erzeugt)* benannt. Sie können dies später in der Adminoberfläche ändern (siehe *Ändern von Sprachdefinitionen*) und es im Issue Tracker melden (siehe *Zu Weblate beitragen*), damit die richtige Definition in die nächste Weblate-Version aufgenommen werden kann.

Hinweis: Falls Sie etwas Unerwünschtes als Sprache sehen, sollten Sie den *Sprachen-Filter* so anpassen, dass solche Dateien beim Parsen von Übersetzungen ignoriert werden.

Siehe auch:

Sprachcode, Neue Übersetzungen hinzufügen

2.8.3 Ändern von Sprachdefinitionen

Sie können Sprachdefinitionen in der Sprachoberfläche (`/languages/` URL) ändern.

Achten Sie bei der Bearbeitung darauf, dass alle Felder korrekt ausgefüllt sind (insbesondere die Pluralformen und die Leserichtung), da die Übersetzer die Übersetzungen sonst nicht richtig bearbeiten können.

2.8.4 Mehrdeutige Sprachcodes und Makrosprachen

In vielen Fällen ist es keine gute Idee, einen Makrosprachcode für eine Übersetzung zu verwenden. Ein typischer Problemfall ist die kurdische Sprache, die je nach Variante in arabischer oder lateinischer Schrift geschrieben werden kann. Um ein korrektes Verhalten in Weblate zu erreichen, wird empfohlen, nur einzelne Sprachcodes zu verwenden und Makrosprachen zu vermeiden.

Siehe auch:

Makrosprachen-Definition, Liste der Makrosprachen

2.8.5 Sprachdefinitionen

Jede Sprache besteht aus den folgenden Feldern:

Sprachcode

Code zur Identifizierung der Sprache. Weblate bevorzugt Zwei-Buchstaben-Codes, wie sie in [ISO 639-1](#) definiert sind, verwendet aber [ISO 639-2](#) oder [ISO 639-3](#) Codes für Sprachen, die keinen Zwei-Buchstaben-Code haben. Es kann auch erweiterte Codes unterstützen, wie in [BCP 47](#) definiert.

Siehe auch:

Parsing von Sprachcodes, Neue Übersetzungen hinzufügen

Sprachenname

Sichtbarer Name der Sprache. Die in Weblate enthaltenen Sprachennamen werden auch je nach Sprache der Benutzeroberfläche lokalisiert.

Leserichtung

Bestimmt, ob die Sprache von rechts nach links oder von links nach rechts geschrieben wird. Diese Eigenschaft wird für die meisten Sprachen automatisch richtig erkannt.

Anzahl der Pluralformen

Anzahl der in der Sprache verwendeten Pluralformen.

Pluralformel

Gettext-kompatible Pluralformel, die verwendet wird, um zu bestimmen, welche Pluralform für eine bestimmte Zählung verwendet wird.

Siehe auch:

Plural, GNU gettext utilities: Plural forms, Language Plural Rules by the Unicode Consortium

Anzahl der Sprecher

Anzahl der weltweiten Sprecher dieser Sprache.

2.8.6 Neue Übersetzungen hinzufügen

Geändert in Version 2.18: In Versionen vor 2.18 war das Verhalten beim Hinzufügen neuer Übersetzungen dateiformatspezifisch.

Weblate kann automatisch eine neue Übersetzung für alle Dateiformate starten.

Some formats expect to start with an empty file and only translated strings to be included (for example *Android String-Ressourcen*), while others expect to have all keys present (for example *GNU gettext*). The document-based formats (for example *OpenDocument-Format*) start with a copy of the source document and all strings marked as needing editing. In some situations this really doesn't depend on the format, but rather on the framework you use to handle the translation (for example with *JSON-Dateien*).

Wenn Sie *Vorlage für neue Übersetzungen* in *Komponentenkonfiguration* angeben, wird Weblate diese Datei verwenden, um neue Übersetzungen zu beginnen. Alle bereits vorhandenen Übersetzungen werden dabei aus der Datei entfernt.

Wenn *Vorlage für neue Übersetzungen* leer ist und das Dateiformat dies unterstützt, wird eine leere Datei erstellt, in die neue Zeichenketten eingefügt werden, sobald sie übersetzt sind.

Mit *Stil des Sprachcodes* können Sie den in den erzeugten Dateinamen verwendeten Sprachcode anpassen:

Standard basierend auf dem Dateiformat

Abhängig vom Dateiformat, für die meisten davon wird POSIX verwendet.

POSIX-Stil mit Unterstrich als Trennzeichen

Wird in der Regel von gettext und ähnlichen Tools verwendet, erzeugt Sprachcodes wie `pt_BR`.

POSIX-Stil mit Unterstrich als Trennzeichen, einschließlich Ländercode

Sprachcode im POSIX-Stil, einschließlich des Ländercodes, auch wenn dies nicht erforderlich ist (z. B. `'cs_CZ'`).

BCP-Stil mit Bindestrich als Trennzeichen

Wird in der Regel auf Webplattformen verwendet und erzeugt Sprachcodes wie `pt-BR`.

BCP-Stil mit Bindestrich als Trennzeichen, einschließlich Ländercode

Sprachcode im BCP-Stil, einschließlich des Ländercodes, auch wenn dies nicht erforderlich ist (z. B. `'cs_CZ'`).

BCP-Stil mit Bindestrich als Trennzeichen, herkömmliche Sprachcodes

Verwendet herkömmliche Codes für die chinesische und die BCP-Schreibweise.

BCP-Stil mit Bindestrich als Trennzeichen, Kleinbuchstaben

BCP-Schreibweise, alles in Kleinbuchstaben (zum Beispiel `cs-cz`).

Apple App Store Metadaten-Stil

Stil, der zum Hochladen von Metadaten in den Apple App Store geeignet ist.

Google Play Metadaten-Stil

Stil, der für das Hochladen von Metadaten in den Google Play Store geeignet ist.

Android-Stil

Wird nur in Android-Anwendungen verwendet, erzeugt Sprachcodes wie `pt-rBR`.

Linux-Stil

Gebietsschemas, wie sie von Linux verwendet werden, verwenden herkömmliche Codes für die chinesische und POSIX-Notation.

Additionally, any mappings defined in *Sprachaliasnamen* are applied in reverse.

Bemerkung: Weblate recognizes any of these when parsing translation files, the above settings only influences how new files are created.

Siehe auch:

Sprachcode, Sprachaliasnamen, Parsing von Sprachcodes

2.9 Kontinuierliche Lokalisierung

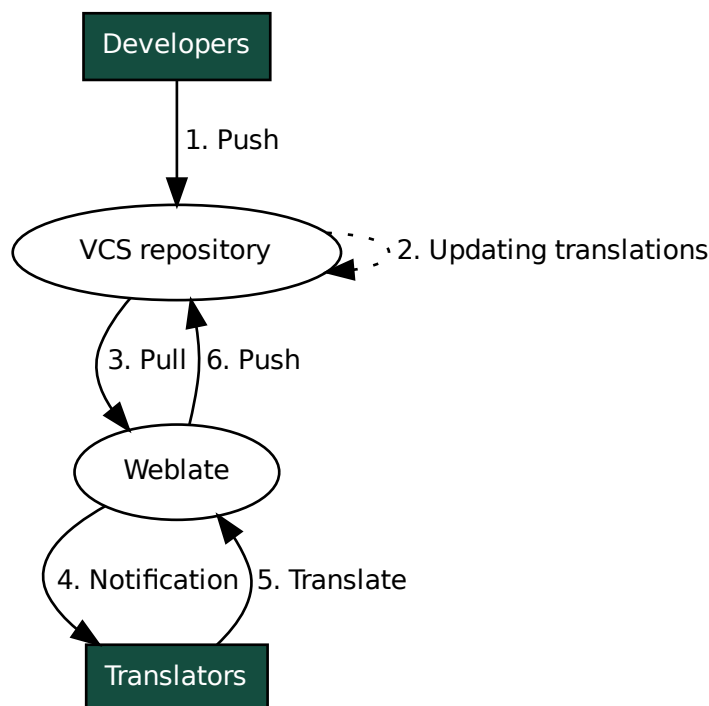
Es gibt eine Infrastruktur, die dafür sorgt, dass Ihre Übersetzung genau der Entwicklung folgt. Auf diese Weise können die Übersetzer die ganze Zeit an den Übersetzungen arbeiten, anstatt kurz vor der Veröffentlichung eine große Menge an neuem Text durchzuarbeiten.

Siehe auch:

`/devel/integration` beschreibt grundlegende Möglichkeiten zur Integration Ihrer Entwicklung mit Weblate.

Dies ist der Prozess:

1. Die Entwickler nehmen Änderungen vor und pushen sie an das VCS-Repository.
2. Optional werden die Übersetzungsdateien aktualisiert (dies hängt vom Dateiformat ab, siehe [Warum zeigt Weblate immer noch alte Übersetzungszeichenketten an, obwohl die Vorlage aktualisiert wurde?](#)).
3. Weblate zieht die Änderungen aus dem VCS-Repository, siehe [Repositorys werden aktualisiert](#).
4. Sobald Weblate Änderungen an Übersetzungen feststellt, werden die Übersetzer entsprechend ihrer Abonnementeinstellungen benachrichtigt.
5. Übersetzer übermitteln ihre Übersetzungen über die Weblate-Weboberfläche oder laden Änderungen offline hoch.
6. Sobald die Übersetzer fertig sind, überträgt Weblate die Änderungen in das lokale Repository (siehe [Lazy Commits](#)) und pushed sie zurück, wenn es dazu berechtigt ist (siehe [Pushen von Änderungen aus Weblate](#)).



2.9.1 Repositorys werden aktualisiert

Sie sollten eine Möglichkeit einrichten, Backend-Repositorys von ihrer Quelle aus zu aktualisieren.

- Verwenden Sie *Benachrichtigungs-Hooks* zur Integration mit den meisten gängigen Code-Hosting-Diensten:
 - *Automatisches Empfangen von Änderungen von GitHub*
 - *Automatischer Empfang von Änderungen von GitLab*
 - *Automatischer Empfang von Änderungen von Bitbucket*
 - *Automatischer Empfang von Änderungen von Pagure*
 - *Automatischer Empfang von Änderungen von Azure Repos*
 - *Automatischer Empfang von Änderungen von Gitea Repos*
- Manuelles Auslösen der Aktualisierung entweder in der Repository-Verwaltung oder mit *Weblates REST API* oder *Weblate-Client*
- Aktivieren Sie `AUTO_UPDATE`, um alle Komponenten in Ihrer Weblate-Installation automatisch zu aktualisieren
- Führen Sie `updategit` aus (mit Auswahl des Projekts oder `--all`, um alle zu aktualisieren)

Wann immer Weblate das Repository aktualisiert, werden die Post-Update-Erweiterungen ausgelöst, siehe *Erweiterungen*.

Vermeiden von Merge-Konflikten

Die Merge-Konflikte von Weblate entstehen, wenn dieselbe Datei sowohl in Weblate als auch außerhalb von Weblate geändert wurde. Es gibt zwei Möglichkeiten, damit umzugehen: Entweder Sie vermeiden Bearbeitungen außerhalb von Weblate oder Sie integrieren Weblate in Ihren Aktualisierungsprozess, so dass die Änderungen vor der Aktualisierung der Dateien außerhalb von Weblate geleert werden.

Der erste Ansatz ist bei einsprachigen Dateien einfach – Sie können neue Zeichenketten innerhalb von Weblate hinzufügen und die gesamte Bearbeitung der Dateien dort belassen. Für zweisprachige Dateien gibt es normalerweise eine Art von Nachrichtenextraktionsprozess, um übersetzbare Dateien aus dem Quellcode zu erzeugen. In manchen Fällen kann dies in zwei Teile aufgeteilt werden – einer für die Extraktion erzeugt eine Vorlage (z. B. gettext POT wird mit `xgettext` erzeugt) und ein weiterer Prozess fügt sie in die tatsächlichen Übersetzungen ein (die gettext PO-Dateien werden mit `msgmerge` aktualisiert). Sie können den zweiten Schritt innerhalb von Weblate durchführen. Weblate sorgt dafür, dass alle anstehenden Änderungen vor diesem Vorgang berücksichtigt werden.

Der zweite Ansatz kann erreicht werden, indem man *Weblates REST API* verwendet, um Weblate zu zwingen, alle anstehenden Änderungen zu pushen und die Übersetzung zu sperren, während man selbst Änderungen vornimmt.

Das Skript zur Durchführung von Aktualisierungen kann wie folgt aussehen:

```
# Lock Weblate translation
wlc lock
# Push changes from Weblate to upstream repository
wlc push
# Pull changes from upstream repository to your local copy
git pull
# Update translation files, this example is for Django
./manage.py makemessages --keep-pot -a
git commit -m 'Locale updates' -- locale
# Push changes to upstream repository
git push
# Tell Weblate to pull changes (not needed if Weblate follows your repo
# automatically)
wlc pull
# Unlock translations
wlc unlock
```


Wenn mehrere Komponenten dasselbe Repository teilen, müssen Sie sie alle separat sperren:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Bemerkung: Das Beispiel verwendet *Weblate-Client*, das eine Konfiguration (API-Schlüssel) benötigt, um Weblate aus der Ferne steuern zu können. Sie können dies auch mit einem beliebigen HTTP-Client anstelle von wlc erreichen, z. B. curl, siehe *Weblates REST API*.

Siehe auch:

Weblate-Client

Automatisches Empfangen von Änderungen von GitHub

Weblate bietet native Unterstützung für GitHub.

Wenn Sie Hosted Weblate verwenden, empfiehlt es sich, die *Weblate-App* zu installieren, damit Sie die korrekte Einrichtung erhalten, ohne viel einrichten zu müssen. Sie kann auch zum Zurückschieben von Änderungen verwendet werden.

Um Benachrichtigungen über jeden Push an ein GitHub-Repository zu erhalten, fügen Sie den Weblate-Webhook in den Repository-Einstellungen hinzu (*Webhooks*), wie auf dem Bild unten gezeigt:

The screenshot shows the GitHub repository settings for 'WeblateOrg / hello'. The 'Webhooks' tab is selected. The 'Add webhook' form is displayed with the following details:

- Payload URL:** `https://hosted.weblate.org/hooks/github/`
- Content type:** `application/x-www-form-urlencoded`
- Secret:** (empty field)
- SSL verification:** A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' with a red button to 'Disable SSL verification'.
- Which events would you like to trigger this webhook?**
 - ☒ Just the push event.
 - ☐ Send me everything.
 - ☐ Let me select individual events.
- Active:** ☒ We will deliver event details when this hook is triggered.
- Add webhook:** (green button)

Für die Nutzdaten-URL fügen Sie `/hooks/github/` an Ihre Weblate-URL an, zum Beispiel für den Hosted Weblate-Dienst ist dies `https://hosted.weblate.org/hooks/github/`.

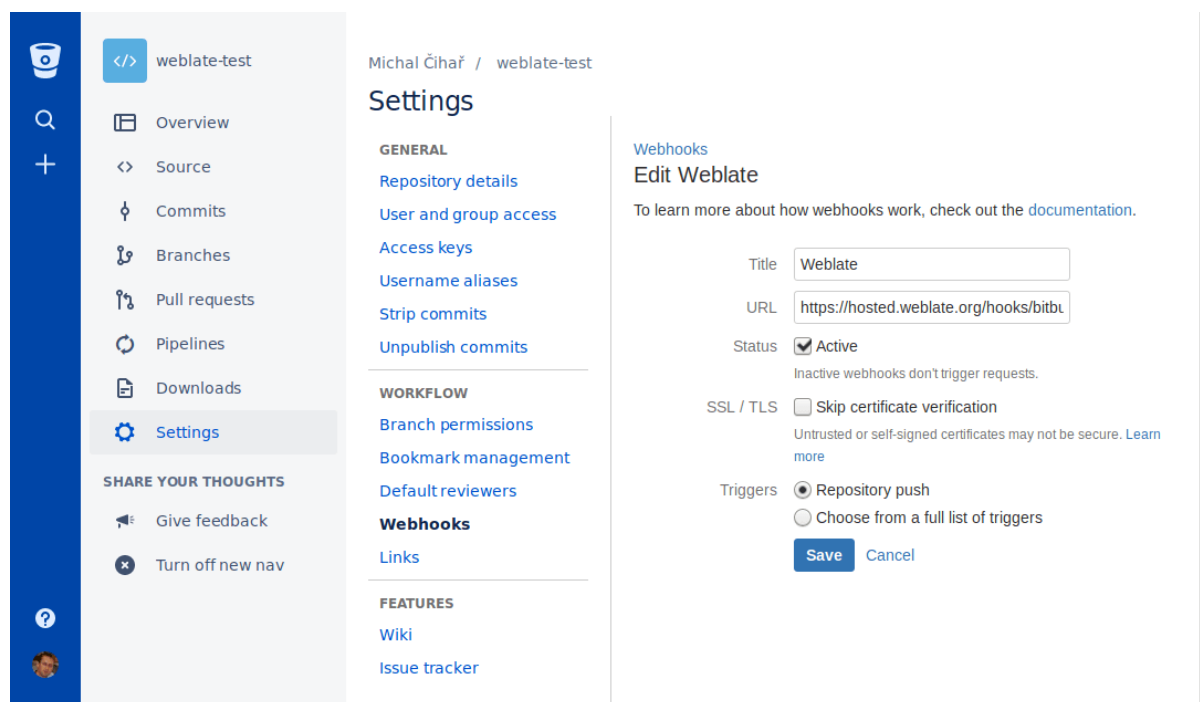
Sie können die anderen Werte auf den Standardeinstellungen belassen (Weblate kann beide Inhaltstypen verarbeiten und verwendet nur das *push*-Ereignis).

Siehe auch:

POST /hooks/github/, Zugriff auf Repositorys von Hosted Weblate aus

Automatischer Empfang von Änderungen von Bitbucket

Weblate unterstützt Bitbucket-Webhooks. Fügen Sie einen Webhook, der bei einem Repository-Push ausgelöst wird, mit der Ziel-URL `/hooks/bitbucket/` auf Ihrer Weblate-Installation hinzu (zum Beispiel `https://hosted.weblate.org/hooks/bitbucket/`).



Siehe auch:

POST /hooks/bitbucket/, Zugriff auf Repositorys von Hosted Weblate aus

Automatischer Empfang von Änderungen von GitLab

Weblate unterstützt GitLab-Hooks. Fügen Sie einen Projekt-Webhook mit der Ziel-URL `/hooks/gitlab/` auf Ihrer Weblate-Installation hinzu (zum Beispiel `https://hosted.weblate.org/hooks/gitlab/`).

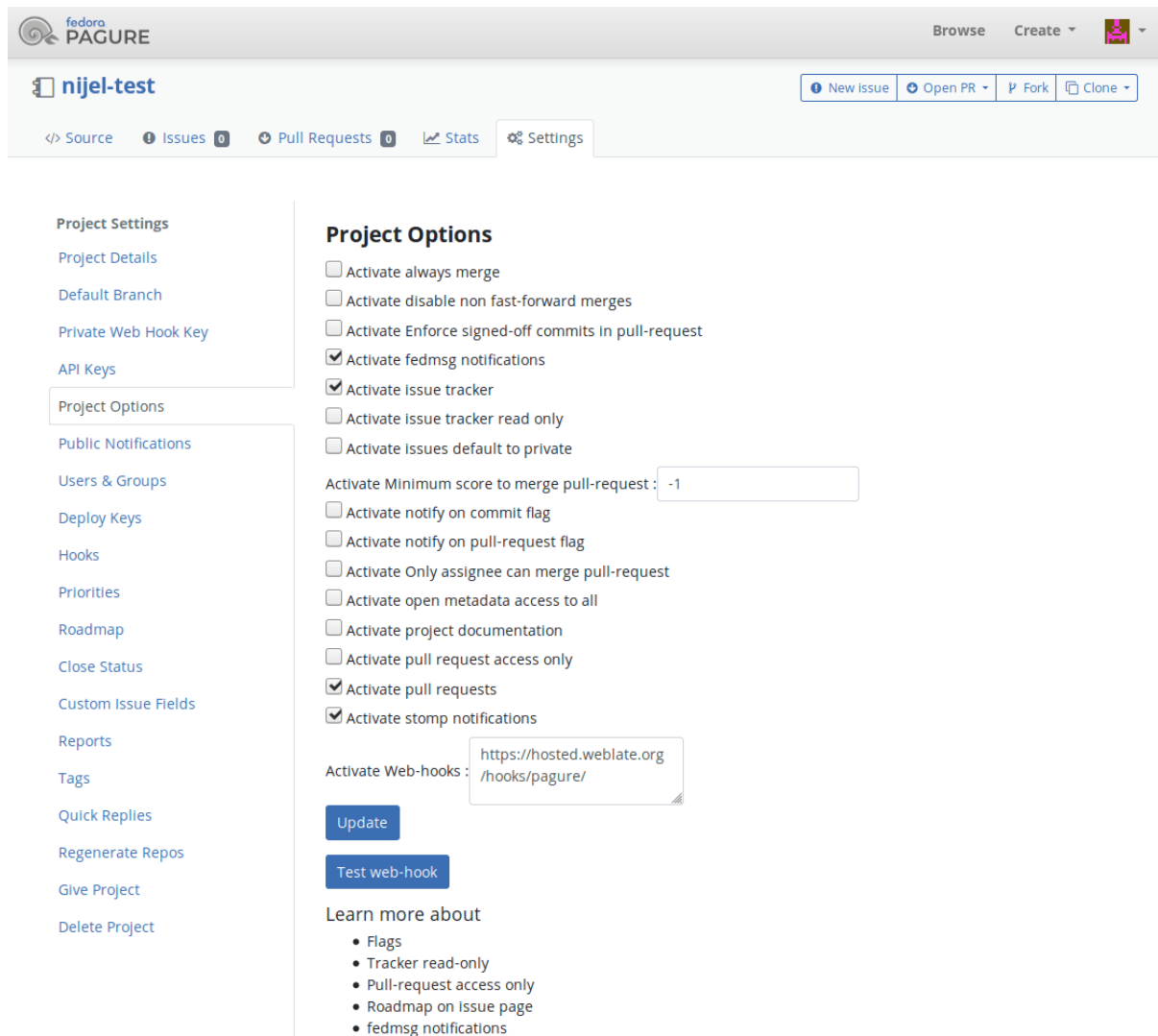
Siehe auch:

POST /hooks/gitlab/, Zugriff auf Repositorys von Hosted Weblate aus

Automatischer Empfang von Änderungen von Pagure

Neu in Version 3.3.

Weblate unterstützt Pagure-Hooks. Fügen Sie einen Webhook mit der Ziel-URL `/hooks/pagure/` auf Ihrer Weblate-Installation hinzu (zum Beispiel `https://hosted.weblate.org/hooks/pagure/`). Dies kann in *Activate Web-hooks* unter *Project options* erfolgen:



The screenshot shows the 'Project Options' section of the Pagure web interface for a project named 'nijel-test'. The interface includes a sidebar with navigation links and a main content area with various settings.

Project Options

- ☐ Activate always merge
- ☐ Activate disable non fast-forward merges
- ☐ Activate Enforce signed-off commits in pull-request
- ☒ Activate fedmsg notifications
- ☒ Activate Issue tracker
- ☐ Activate Issue tracker read only
- ☐ Activate Issues default to private
- Activate Minimum score to merge pull-request :
- ☐ Activate notify on commit flag
- ☐ Activate notify on pull-request flag
- ☐ Activate Only assignee can merge pull-request
- ☐ Activate open metadata access to all
- ☐ Activate project documentation
- ☐ Activate pull request access only
- ☒ Activate pull requests
- ☒ Activate stomp notifications

Activate Web-hooks :

Learn more about

- Flags
- Tracker read-only
- Pull-request access only
- Roadmap on Issue page
- fedmsg notifications

Siehe auch:

POST /hooks/pagure/, Zugriff auf Repositorys von Hosted Weblate aus

Automatischer Empfang von Änderungen von Azure Repos

Neu in Version 3.8.

Weblate unterstützt Azure-Repos-Webhooks. Fügen Sie einen Webhook für das *Code pushed*-Ereignis mit der Ziel-URL `/hooks/azure/` auf Ihrer Weblate-Installation hinzu (zum Beispiel `https://hosted.weblate.org/hooks/azure/`). Dies kann in *Service hooks* unter *Project settings* erfolgen.

Siehe auch:

Webhooks im Azure-DevOps-Handbuch, *POST /hooks/azure/*, *Zugriff auf Repositorys von Hosted Weblate aus*

Automatischer Empfang von Änderungen von Gitea Repos

Neu in Version 3.9.

Weblate unterstützt für Gitea-Webhooks. Fügen Sie einen *Gitea Webhook* für das *Push events*-Ereignis mit der Ziel-URL `/hooks/gitea/` auf Ihrer Weblate-Installation hinzu (zum Beispiel `https://hosted.weblate.org/hooks/gitea/`). Dies kann in *Webhooks* unter *Repository Settings* erfolgen.

Siehe auch:

Webhooks im Gitea-Handbuch, *POST /hooks/gitea/*, *Zugriff auf Repositorys von Hosted Weblate aus*

Automatischer Empfang von Änderungen von Gitee Repos

Neu in Version 3.9.

Weblate unterstützt Gitee-Webhooks. Fügen Sie einen *WebHook* für das *Push*-Ereignis mit der Ziel-URL `/hooks/gitee/` auf Ihrer Weblate-Installation hinzu (zum Beispiel `https://hosted.weblate.org/hooks/gitee/`). Dies kann in *WebHooks* unter *Repository Management* erfolgen.

Siehe auch:

Webhooks im Gitee-Handbuch, *POST /hooks/gitee/*, *Zugriff auf Repositorys von Hosted Weblate aus*

Automatische nächtliche Aktualisierung der Repositorys

Weblate holt sich nachts automatisch entfernte Repositorys, um die Leistung beim späteren Zusammenführen von Änderungen zu verbessern. Sie können dies optional auch in nächtliche Zusammenführungen umwandeln, indem Sie *AUTO_UPDATE* aktivieren.

2.9.2 Pushen von Änderungen aus Weblate

Für jede Übersetzungskomponente kann eine Push-URL eingerichtet werden (siehe *Push-URL für Repository*) und in diesem Fall kann Weblate Änderungen an das entfernte Repository weiterleiten. Weblate kann auch so konfiguriert werden, dass Änderungen automatisch bei jedem Commit gepusht werden (dies ist die Voreinstellung, siehe *Bei Commit gleichzeitig Pushen*). Wenn Sie nicht möchten, dass Änderungen automatisch gepusht werden, können Sie dies manuell unter *Paketquellwartung* oder über die API per *wlc push* tun.

Die Push-Optionen unterscheiden sich je nach dem verwendeten *Integration der Versionsverwaltung*, weitere Einzelheiten finden Sie in diesem Kapitel.

Für den Fall, dass Sie keine direkten Pushes durch Weblate wünschen, gibt es Unterstützung für Pull Requests von *GitHub-Pull-Requests*, *GitLab-Merge-Requests*, *Gitea-Pull-Requests*, *Pagure-Merge-Requests* oder *Gerrit-Reviews*. Sie können diese aktivieren, indem Sie *GitHub*, *GitLab*, *Gitea*, *Gerrit* oder *Pagure* als *Versionsverwaltung* in *Komponentenkonfiguration* wählen.

Insgesamt stehen mit Git, GitHub und GitLab die folgenden Optionen zur Verfügung:

Gewünschte Einrichtung	Versionsverwaltung	Push-URL für Repository	Push Branch
Kein Push	<i>Git</i>	<i>leer</i>	<i>leer</i>
Direkt pushen	<i>Git</i>	SSH-URL	<i>leer</i>
Push zu separatem Branch	<i>Git</i>	SSH-URL	Name des Branches
GitHub-Pull-Request vom Fork	<i>GitHub-Pull-Requests</i>	<i>leer</i>	<i>leer</i>
GitHub-Pull-Request vom Branch	<i>GitHub-Pull-Requests</i>	SSH-URL ¹	Name des Branches
GitLab-Merge-Request vom Fork	<i>GitLab-Merge-Requests</i>	<i>leer</i>	<i>leer</i>
GitLab-Merge-Request vom Branch	<i>GitLab-Merge-Requests</i>	SSH-URL ^{Seite 297, 1}	Name des Branches
Gitea-Merge-Request vom Fork	<i>Gitea-Pull-Requests</i>	<i>leer</i>	<i>leer</i>
Gitea-Merge-Request vom Branch	<i>Gitea-Pull-Requests</i>	SSH-URL ¹	Name des Branches
Pagure-Merge-Request vom Fork	<i>Pagure-Merge-Requests</i>	<i>leer</i>	<i>leer</i>
Pagure-Merge-Request vom Branch	<i>Pagure-Merge-Requests</i>	SSH-URL ¹	Name des Branches

Bemerkung: Sie können auch das automatische Pushen von Änderungen nach Weblate-Commits aktivieren, dies kann in *Bei Commit gleichzeitig Pushen* erfolgen.

Siehe auch:

Siehe *Zugriff auf Repositories* für die Einrichtung von SSH-Schlüsseln und *Lazy Commits* für Informationen darüber, wann Weblate entscheidet, Änderungen zu committen.

Geschützte Branches

Wenn Sie Weblate auf einem geschützten Branch verwenden, können Sie es so konfigurieren, dass es Pull Requests verwendet und die Übersetzungen tatsächlich überprüft (was bei Sprachen, die Sie nicht kennen, problematisch sein könnte). Ein alternativer Ansatz besteht darin, diese Einschränkung für den Weblate-Push-Nutzer aufzuheben.

Auf GitHub kann dies zum Beispiel in der Repository-Konfiguration erfolgen:

¹ Kann leer sein, falls das *Quellcode-Repository* Pushen unterstützt.

☒ **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

Required approving reviews: 1 ▾

☐ **Dismiss stale pull request approvals when new commits are pushed**

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

☐ **Require review from Code Owners**

Require an approved review in pull requests including files with a designated code owner.

☒ **Restrict who can dismiss pull request reviews**

Specify people or teams allowed to dismiss pull request reviews.

🔍 Search for people or teams

People and teams that can dismiss reviews.**Organization and repository administrators**

These members can always dismiss.

**weblate**

Weblate push user



2.9.3 Mit anderen interagieren

Weblate macht es einfach, mit anderen über seine API zu interagieren.

Siehe auch:

[Weblates REST API](#)

2.9.4 Lazy Commits

Weblate fasst Commits desselben Autors nach Möglichkeit in einem Commit zusammen. Dadurch wird die Anzahl der Commits stark reduziert, allerdings müssen Sie es eventuell explizit anweisen, die Commits durchzuführen, wenn Sie das VCS-Repository synchronisieren wollen, zum Beispiel für Merge (dies ist standardmäßig für die Gruppe *Manager* erlaubt, siehe *[Liste der Berechtigungen und integrierten Rollen](#)*).

Die Änderungen in diesem Modus werden übernommen, sobald eine der folgenden Bedingungen erfüllt ist:

- Jemand anderes ändert eine bereits geänderte Zeichenkette.
- Es erfolgt ein Merge von Upstream.
- Ein expliziter Commit wird angefordert.
- Ein Dateidownload ist angefordert.
- Die Änderung ist älter als der als *Alter der Änderungen, das erreicht sein muss, bevor ein Commit erfolgt* definierte Zeitraum auf *[Komponentenkonfiguration](#)*.

Hinweis: Commits werden für jede Komponente erstellt. Wenn Sie also viele Komponenten haben, werden Sie trotzdem viele Commits sehen. In diesem Fall können Sie die Erweiterung *[Git-Commits zusammenfassen](#)* verwenden.

If you want to commit changes more frequently and without checking of age, you can schedule a regular task to perform a commit. This can be done using *Periodic Tasks* in *Die Django-Adminoberfläche*. First create desired *Interval* (for example 120 seconds). Then add new periodic task and choose `weblate.trans.tasks.commit_pending` as *Task* with `{"hours": 0}` as *Keyword Arguments* and desired interval.

2.9.5 Repository mit Skripten verarbeiten

Die Art und Weise, wie Weblate mit dem Repository interagiert, kann mit *Erweiterungen* angepasst werden. Konsultieren Sie *Ausführen von Skripten der Erweiterung* für Informationen darüber, wie man externe Skripte durch Erweiterungen ausführt.

2.9.6 Übersetzungen aller Komponenten gleich halten

Wenn Sie mehrere Übersetzungskomponenten haben, möchten Sie möglicherweise sicherstellen, dass die selben Zeichenketten dieselbe Übersetzung haben. Dies kann auf mehreren Ebenen erreicht werden.

Übersetzungsweitergabe

Wenn *Verbreitung von Übersetzungen erlauben* aktiviert ist (was die Voreinstellung ist, siehe *Komponentenkonfiguration*), werden alle neuen Übersetzungen automatisch in allen Komponenten mit übereinstimmenden Zeichenketten durchgeführt. Solche Übersetzungen werden dem aktuell übersetzenden Benutzer in allen Komponenten korrekt zugeordnet.

Bemerkung: Die Übersetzungsweitergabe erfordert, dass der Schlüssel mit einsprachigen Übersetzungsformaten übereinstimmt, daher sollten Sie dies bei der Erstellung von Übersetzungsschlüsseln berücksichtigen.

Konsistenzprüfung

Die Prüfung *Inkonsistent* wird immer dann ausgelöst, wenn die Zeichenketten unterschiedlich sind. Sie können dies nutzen, um solche Unterschiede manuell zu überprüfen und die richtige Übersetzung zu wählen.

Automatische Übersetzung

Die automatische Übersetzung basierend auf verschiedenen Komponenten kann eine Möglichkeit sein, die Übersetzungen zwischen den Komponenten zu synchronisieren. Sie können es entweder manuell auslösen (siehe *Automatische Übersetzung*) oder es automatisch bei der Aktualisierung des Repositories mit Hilfe einer Erweiterung laufen lassen (siehe *Automatische Übersetzung*).

2.10 Licensing translations

You can specify which license translations are contributed under. This is especially important to do if translations are open to the public, to stipulate what they can be used for.

You should specify *Komponentenkonfiguration* license info. You should avoid requiring a contributor license agreement, though it is possible.

2.10.1 License info

Upon specifying license info (license name and URL), this info is shown in the translation info section of the respective *Komponentenkonfiguration*.

Usually this is best place to post licensing info if no explicit consent is required. If your project or translation is not libre you most probably need prior consent.

2.10.2 Mitwirkungsvereinbarung

If you specify a contributor license agreement, only users who have agreed to it will be able to contribute. This is a clearly visible step when accessing the translation:

Contribution to this translation requires you to agree with a contributor agreement. [View contributor agreement](#)

Language	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
Czech 🇨🇪 GPL-3.0	✓						
Hebrew 🇮🇱 GPL-3.0	✓						
Hungarian 🇭🇺 GPL-3.0	81%	4	5	32			
English 🇬🇧 GPL-3.0	✓						

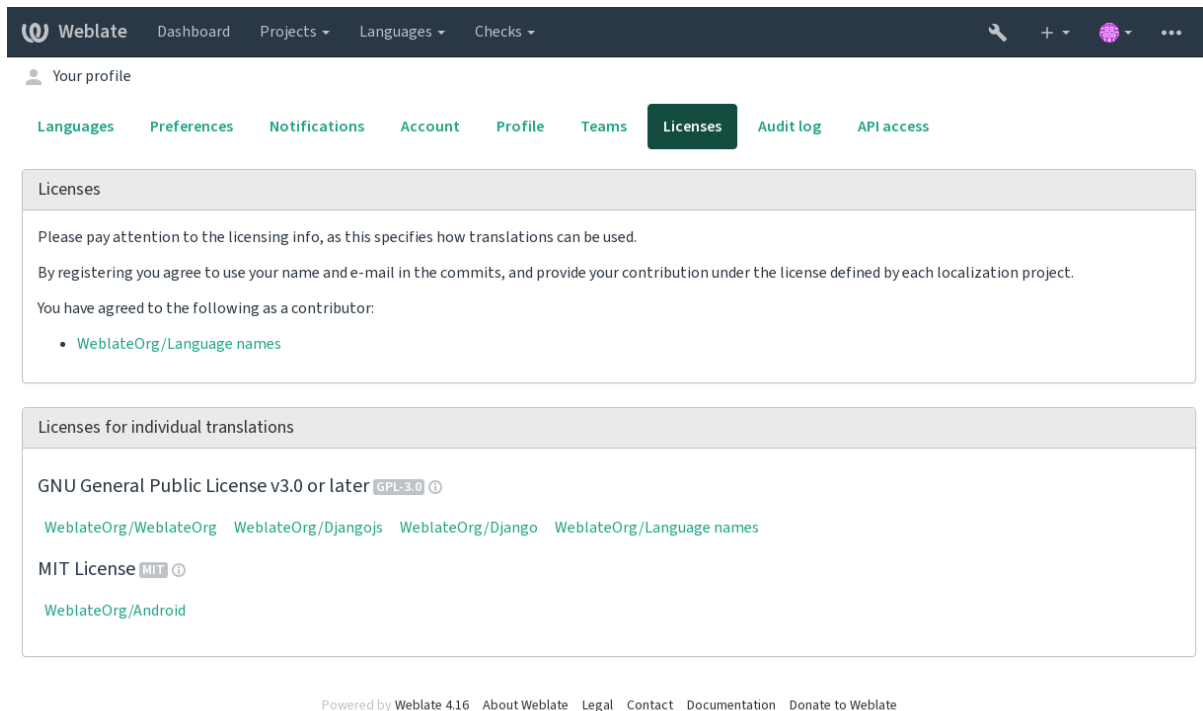
[Start new translation](#)

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

The entered text is formatted into paragraphs and external links can be included. HTML markup can not be used.

2.10.3 User licenses

Any user can review all translation licenses of all public projects on the instance from their profile:



The screenshot shows the Weblate web application interface. At the top is a dark navigation bar with the Weblate logo and links for Dashboard, Projects, Languages, and Checks. Below this is a user profile section with a 'Your profile' link and a row of tabs: Languages, Preferences, Notifications, Account, Profile, Teams, Licenses (which is active), Audit log, and API access. The 'Licenses' tab is open, displaying a section titled 'Licenses' with the following text: 'Please pay attention to the licensing info, as this specifies how translations can be used. By registering you agree to use your name and e-mail in the commits, and provide your contribution under the license defined by each localization project. You have agreed to the following as a contributor:'. A bulleted list follows: '• WeblateOrg/Language names'. Below this is a section titled 'Licenses for individual translations' which lists two licenses: 'GNU General Public License v3.0 or later' (with a link to GPL-3.0) and 'MIT License' (with a link to MIT). Under the MIT License, there is a link to 'WeblateOrg/Android'. At the bottom of the page, a footer line reads: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

2.11 Übersetzungsrichtlinien

2.11.1 Abstimmen über Vorschläge

Jeder kann standardmäßig Vorschläge hinzufügen, die von angemeldeten Benutzern akzeptiert werden. Die Vorschlagsabstimmung kann verwendet werden, um eine Zeichenkette automatisch zu verwenden, wenn mehr als ein angemeldeter Benutzer zustimmt. Dies wird unter der *Komponentenkonfiguration* mit *Abstimmen über Vorschläge* eingeschaltet. *Vorschläge automatisch annehmen* setzt einen Schwellenwert für angenommene Vorschläge (dies schließt die Stimme des vorschlagenden Benutzers ein, sofern sie abgegeben wird).

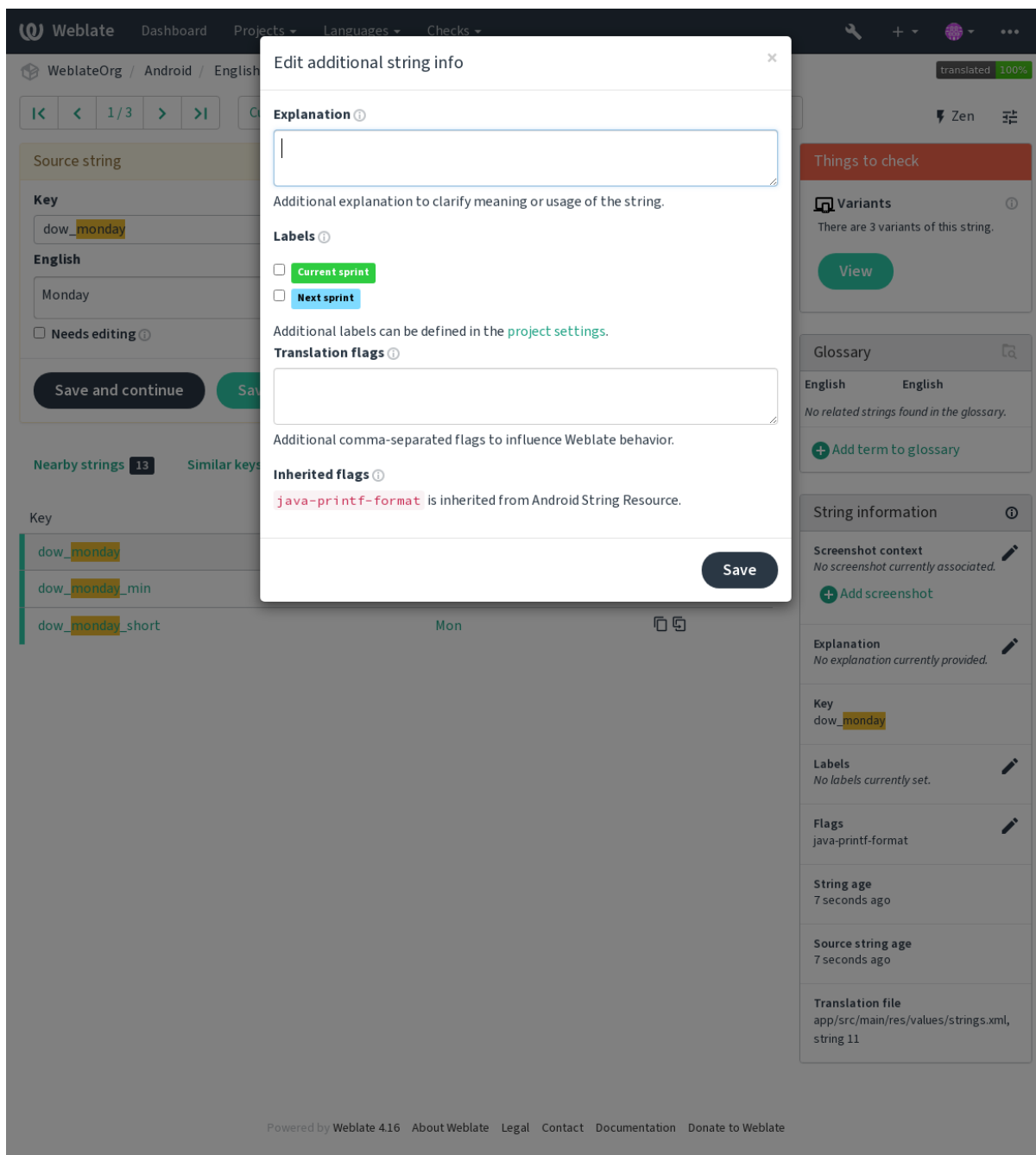
Bemerkung: Once automatic acceptance is set up, normal users lose the privilege to directly save translations or accept suggestions. This can be overridden with the *Edit string when suggestions are enforced permission*.

You can combine these with *access control* into one of the following setups:

- Users suggest and vote for suggestions and a limited group controls what is accepted. - Turn on voting. - Turn off automatic acceptance. - Don't let users save translations.
- Users suggest and vote for suggestions with automatic acceptance once the defined number of them agree. - Turn on voting. - Set the desired number of votes for automatic acceptance.
- Optional voting for suggestions. (Can optionally be used by users when they are unsure about a translation by making multiple suggestions.) - Only turn on voting.

2.11.2 Zusätzliche Informationen über Ausgangszeichenketten

Enhance the translation process by adding additional info to the strings including explanations, string priorities, check flags and visual context. Some of that info may be extracted from the translation files and some may be added by editing the additional string info:



Sie können direkt von der Übersetzungsoberfläche aus darauf zugreifen, indem Sie auf das „Bearbeiten“-Symbol neben *Bildschirmfoto zum Kontext* oder *Markierungen* klicken.

Weblate
 Dashboard
 Projects
 Languages
 Checks

Translated 96%

WeblateOrg / Django / Czech / Translate

All strings

Position and priority

Zen

Translation

Explanation

Help text for automatic translation tool

English

Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

☐ Needs editing

Save and continue Save and stay Suggest Skip

Glossary

English Czech

machine strojový překlád weblate.org

project projekt weblate.org

Add term to glossary

String information

Screenshot contact
No screenshot currently associated.
Add screenshot

Explanation
Help text for automatic translation tool

Labels
No labels currently set.

Flags
No flags currently set.

Source string location
weblate/templates/translation.html:212

String age
2 seconds ago

Source string age
2 seconds ago

Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 11

Nearby strings 26 Comments Automatic suggestions Other languages 4 History

Context	English	Czech	Actions
	Files	Soubory	
	Automatic translation	Automatický překlad	
	Add new translation string	Add new translation string	
	Translation status	Stav překladu	
	Singular %(count)s word Plural %(count)s words	One %(count)s slovo Few %(count)s slova Many %(count)s slov	
	Other components	Další součásti	
	Translation file	Soubor s překladem	
	Download	Stáhnout	
	Browse all translation changes	Procházet všechny změny v překladu.	
	Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.	Automatický překlad použije stávající překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.	
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.	
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahráný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.	
	The uploaded file will be merged with the current translation.	Nahráný soubor bude sloučen se stávajícími překlady.	
	The fulltext search might not work properly as the fulltext index for this translation is not yet up to date.	Fulltextové vyhledávání nemusí fungovat správně, protože fulltextový index pro tento překlad ještě není plně zpracován.	
	Review	Kontrola	
	Review translations touched by other users.	Zkontrolovat překlady od ostatních uživatelů.	
	Start review	Začít kontrolu	
	Percent	Procenta	
	Total	Celkem	
	Failing check	Neúspěšných kontrol	
	Last activity	Poslední aktivity	
	Last change	Poslední změna	
	Last author	Poslední autor	
	What is %s?	Kolik to je?	
	Question for a mathematics-based CAPTCHA, the %s is an arithmetic problem		
	The string uses three dots (...) instead of an ellipsis character (...)		

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Priorisierung von Zeichenketten

Neu in Version 2.0.

String priority can be changed to offer higher priority strings for translation earlier by using the `priority` flag.

Hinweis: This can be used to order the flow of translation in a logical manner.

Siehe auch:

Qualitätsprüfungen

Übersetzungsmarkierungen

Neu in Version 2.4.

Geändert in Version 3.3: Previously called *Quality checks flags*, it no longer configures only checks.

Customization of quality checks and other Weblate behavior, see *Anpassen des Verhaltens mit Markierungen*.

The string flags are also inherited from the *Übersetzungsmarkierungen* at *Komponentenkonfiguration* and flags from the translation file (see *Unterstützte Dateiformate*).

Siehe auch:

Qualitätsprüfungen, Anpassen des Verhaltens mit Markierungen

Erläuterung

Geändert in Version 4.1: In previous versions this has been called *Extra context*.

Use the explanation to clarify scope or usage of the translation. You can use Markdown to include links and other markup.

Visueller Kontext für Zeichenketten

Neu in Version 2.9.

Sie können ein Bildschirmfoto hochladen, das die Verwendung einer bestimmten Ausgangszeichenkette in Ihrem Programm zeigt. Dies hilft den Übersetzern zu verstehen, wo sie verwendet wird und wie sie übersetzt werden sollte.

Das hochgeladene Bildschirmfoto wird in der Seitenleiste des Übersetzungskontexts angezeigt:

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The main translation area shows the English source text: 'Help text for automatic translation tool' and the Czech machine translation: 'Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.' The interface includes navigation buttons like 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. The right sidebar contains a 'Glossary' with terms like 'machine translation' and 'project', a 'String information' panel with details like 'Source string location' and 'String age', and a 'Translation memory' search bar.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

In Ergänzung zu *Zusätzliche Informationen über Ausgangszeichenketten* haben Bildschirmfotos eine eigene Verwaltungsoberfläche unter dem Menü *Werkzeuge*. Laden Sie Bildschirmfotos hoch, ordnen Sie sie manuell den Ausgangszeichenketten zu oder verwenden Sie dazu die optische Zeichenerkennung.

Sobald ein Bildschirmfoto hochgeladen ist, übernimmt diese Schnittstelle die Verwaltung und die Zuordnung der Ausgangszeichenkette:

WebateDashboardProjectsLanguagesChecks

WebateOrg / Django / Screenshots / Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			
Screenshot is shown to add visual context for all listed source strings.			

Assign source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			

Source string search

Search

Automatically recognize

Image

Source string

Hello, world!

OneOrangutan has %d banana.

OtherOrangutan has %d bananas.

Try Weblate at <http://demo.weblate.org/>!

Thank you for using Weblate.

Screenshot is shown to add visual context for all listed source strings.

Edit screenshot

Screenshot name

Automatic translation

Image

Currently: screenshots/screenshot.png

Change:

Choose File

No file chosen

Upload JPEG or PNG images up to 2000x2000 pixels.

Save

Screenshot details

Created	now
Uploaded by	testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

2.12 Überprüfungen und Korrekturen

2.12.1 Benutzerdefinierte automatische Korrekturen

Sie können auch Ihre eigenen automatischen Korrekturen zusätzlich zu den Standardkorrekturen implementieren und sie in `AUTOFIX_LIST` aufnehmen.

Die automatischen Korrekturen sind leistungsstark, können aber auch Schaden anrichten; seien Sie vorsichtig, wenn Sie eine schreiben.

Zum Beispiel würde die folgende automatische Korrektur jedes Vorkommen der Zeichenfolge `foo` in einer Übersetzung durch `bar` ersetzen:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy as _

from weblate.trans.autofixes.base import AutoFix

class ReplaceFooWithBar(AutoFix):
    """Replace foo with bar."""

    name = _("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False
```

Um benutzerdefinierte Prüfungen zu installieren, geben Sie in `AUTOFIX_LIST` einen vollqualifizierten Pfad zu der Python-Klasse an, siehe *Benutzerdefinierte Qualitätsprüfungen, Erweiterungen und automatische Korrekturen*.

2.12.2 Anpassen des Verhaltens mit Markierungen

Sie können das Verhalten von Weblate durch die Verwendung von Markierungen feinabstimmen. Dies kann auf der Ebene der Ausgangszeichenkette geschehen (siehe *Zusätzliche Informationen über Ausgangszeichenketten*), oder in der *Komponentenkonfiguration (Übersetzungsmarkierungen)*. Einige Dateiformate erlauben es auch, Markierungen direkt im Format anzugeben (siehe *Unterstützte Dateiformate*).

Die Markierungen sind durch Kommas getrennt, die Parameter durch Doppelpunkte. Sie können Anführungszeichen verwenden, um Leerzeichen oder Sonderzeichen in die Zeichenkette aufzunehmen. Zum Beispiel:

```
placeholders:"special:value":"other value", regex:.*
```

Es werden sowohl einfache als auch doppelte Anführungszeichen akzeptiert, Sonderzeichen werden mit Backslash maskiert:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

Hier finden Sie eine Liste der derzeit akzeptierten Markierungen:

rst-text

Behandelt einen Text als ein reStructuredText-Dokument, wirkt sich auf *Unveränderte Übersetzung* aus.

dos-eol

Verwendet DOS-Zeilende-Markierungen anstelle von Unix-Markierungen (`\r\n` anstelle von `\n`).

read-only

Die Zeichenkette ist schreibgeschützt und sollte in Weblate nicht bearbeitet werden, siehe [Schreibgeschützte Zeichenketten](#).

priority:N

Priorität der Zeichenkette. Zeichenketten mit höherer Priorität werden zuerst zur Übersetzung angeboten. Die Standardpriorität ist 100. Je höher die Priorität einer Zeichenkette ist, desto früher wird sie zur Übersetzung angeboten.

max-length:N

Begrenzt die maximale Länge für eine Zeichenkette auf N Zeichen, siehe [Maximale Länge der Übersetzung](#).

xml-text

Behandelt Text als XML-Dokument, betrifft [XML-Syntax](#) und [XML-Auszeichnung](#).

font-family:NAME

Definieren Sie die Schriftfamilie für Rendering-Prüfungen, siehe [Schriftarten verwalten](#).

font-weight:WEIGHT

Definieren Sie das Schriftgewicht für Rendering-Prüfungen, siehe [Schriftarten verwalten](#).

font-size:SIZE

Definieren Sie die Schriftgröße für Rendering-Prüfungen, siehe [Schriftarten verwalten](#).

font-spacing:SPACING

Definieren Sie die Buchstabenabstände für Rendering-Prüfungen, siehe [Schriftarten verwalten](#).

icu-flags:FLAGS

Definieren Sie Markierungen, um das Verhalten der Qualitätsprüfung [ICU MessageFormat](#) anzupassen.

icu-tag-prefix:PREFIX

Setzen Sie ein erforderliches Präfix für XML-Tags für die Qualitätsprüfung [ICU MessageFormat](#).

placeholders:NAME:NAME2:...

Platzhalter-Zeichenketten werden in der Übersetzung erwartet, siehe [Platzhalter](#).

replacements:FROM:TO:FROM2:TO2...

Ersetzungen, die bei der Überprüfung der resultierenden Textparameter durchgeführt werden sollen (zum Beispiel in [Maximaler Umfang der Übersetzung](#) oder [Maximale Länge der Übersetzung](#)). Der typische Anwendungsfall hierfür ist die Erweiterung von Platzhaltern, um sicherzustellen, dass der Text auch bei langen Werten passt, zum Beispiel: `replacements:%s:"John Doe"`.

variants:SOURCE

Markiert diese Zeichenkette als Variante der Zeichenkette mit gleicher Ausgangszeichenkette. Siehe [variants](#).

regex:REGEX

Regulärer Ausdruck zum Abgleich mit der Übersetzung, siehe [Regulärer Ausdruck](#).

forbidden

Zeigt eine verbotene Übersetzung in einem Glossar an, siehe [Unzulässige Übersetzungen](#).

strict-same

Unveränderte Übersetzung“ vermeiden, indem man die eingebaute schwarze Liste der Wörter verwendet, siehe [Unveränderte Übersetzung](#).

check-glossary

Aktiviere Qualitätsprüfung [Folgt nicht dem Glossar](#).

angularjs-format

Aktiviere Qualitätsprüfung [AngularJS-Interpolations-Zeichenkette](#).

c-format

Aktiviere Qualitätsprüfung [C-Format](#).

c-sharp-format

Aktiviere Qualitätsprüfung [C#-Format](#).

es-format

Aktiviere Qualitätsprüfung *ECMAScript-Buchstabenvorlagen*.

i18next-interpolation

Aktiviere Qualitätsprüfung *i18next Interpolation*.

icu-message-format

Aktiviere Qualitätsprüfung *ICU MessageFormat*.

java-printf-format

Aktiviere Qualitätsprüfung *Java-Format*.

java-format

Aktiviere Qualitätsprüfung *Java-MessageFormat*.

javascript-format

Aktiviere Qualitätsprüfung *JavaScript-Format*.

lua-format

Aktiviere Qualitätsprüfung *Lua-Format*.

object-pascal-format

Aktiviere Qualitätsprüfung *Object Pascal-Format*.

percent-placeholders

Aktiviere Qualitätsprüfung *Platzhalter in Prozent*.

perl-format

Aktiviere Qualitätsprüfung *Perl-Format*.

php-format

Aktiviere Qualitätsprüfung *PHP-Format*.

python-brace-format

Aktiviere Qualitätsprüfung *Python-Brace-Format*.

python-format

Aktiviere Qualitätsprüfung *Python-Format*.

qt-format

Aktiviere Qualitätsprüfung *Qt-Format*.

qt-plural-format

Aktiviere Qualitätsprüfung *Qt-Plural-Format*.

ruby-format

Aktiviere Qualitätsprüfung *Ruby-Format*.

scheme-format

Aktiviere Qualitätsprüfung *Scheme-Format*.

vue-format

Aktiviere Qualitätsprüfung *Vue I18n-Formatierung*.

md-text

Text wie ein Markdown-Dokument behandeln. Aktivieren Sie die Qualitätsprüfungen *Markdown-Links*, *Markdown-Referenzen*, und *Markdown-Syntax*.

case-insensitive

Anpassung des Prüfverhaltens, so dass Groß- und Kleinschreibung nicht berücksichtigt werden. Betrifft derzeit nur die Qualitätsprüfung *Platzhalter*.

safe-html

Aktiviere Qualitätsprüfung *Unsicheres HTML*.

url

Die Zeichenkette sollte nur aus einer URL bestehen. Aktivieren Sie die Qualitätsprüfung *URL*.

ignore-all-checks

Ignoriere alle Qualitätsprüfungen.

ignore-bbcode

Überspringe Qualitätsprüfung *BBCode-Markup*.

ignore-duplicate

Überspringe Qualitätsprüfung *Aufeinanderfolgende doppelte Wörter*.

ignore-check-glossary

Überspringe Qualitätsprüfung *Folgt nicht dem Glossar*.

ignore-double-space

Überspringe Qualitätsprüfung *Doppeltes Leerzeichen*.

ignore-angularjs-format

Überspringe Qualitätsprüfung *AngularJS-Interpolations-Zeichenkette*.

ignore-c-format

Überspringe Qualitätsprüfung *C-Format*.

ignore-c-sharp-format

Überspringe Qualitätsprüfung *C#-Format*.

ignore-es-format

Überspringe Qualitätsprüfung *ECMAScript-Buchstabenvorlagen*.

ignore-i18next-interpolation

Überspringe Qualitätsprüfung *i18next Interpolation*.

ignore-icu-message-format

Überspringe Qualitätsprüfung *ICU MessageFormat*.

ignore-java-format

Überspringe Qualitätsprüfung *Java-MessageFormat*.

ignore-java-printf-format

Überspringe Qualitätsprüfung *Java-Format*.

ignore-javascript-format

Überspringe Qualitätsprüfung *JavaScript-Format*.

ignore-lua-format

Überspringe Qualitätsprüfung *Lua-Format*.

ignore-object-pascal-format

Überspringe Qualitätsprüfung *Object Pascal-Format*.

ignore-percent-placeholders

Überspringe Qualitätsprüfung *Platzhalter in Prozent*.

ignore-perl-format

Überspringe Qualitätsprüfung *Perl-Format*.

ignore-php-format

Überspringe Qualitätsprüfung *PHP-Format*.

ignore-python-brace-format

Überspringe Qualitätsprüfung *Python-Brace-Format*.

ignore-python-format

Überspringe Qualitätsprüfung *Python-Format*.

ignore-qt-format

Überspringe Qualitätsprüfung *Qt-Format*.

ignore-qt-plural-format

Überspringe Qualitätsprüfung *Qt-Plural-Format*.

ignore-ruby-format

Überspringe Qualitätsprüfung *Ruby-Format*.

ignore-scheme-format

Überspringe Qualitätsprüfung *Scheme-Format*.

ignore-vue-format

Überspringe Qualitätsprüfung *Vue I18n-Formatierung*.

ignore-translated

Überspringe Qualitätsprüfung *Ist übersetzt worden*.

ignore-inconsistent

Überspringe Qualitätsprüfung *Inkonsistent*.

ignore-kashida

Überspringe Qualitätsprüfung *Kashida-Buchstabe verwendet*.

ignore-md-link

Überspringe Qualitätsprüfung *Markdown-Links*.

ignore-md-reflink

Überspringe Qualitätsprüfung *Markdown-Referenzen*.

ignore-md-syntax

Überspringe Qualitätsprüfung *Markdown-Syntax*.

ignore-max-length

Überspringe Qualitätsprüfung *Maximale Länge der Übersetzung*.

ignore-max-size

Überspringe Qualitätsprüfung *Maximaler Umfang der Übersetzung*.

ignore-escaped-newline

Überspringe Qualitätsprüfung *Unterschiedliche Menge \n*.

ignore-end-colon

Überspringe Qualitätsprüfung *Nicht übereinstimmender Doppelpunkt*.

ignore-end-ellipsis

Überspringe Qualitätsprüfung *Nicht übereinstimmende Auslassungspunkte*.

ignore-end-exclamation

Überspringe Qualitätsprüfung *Nicht übereinstimmendes Ausrufezeichen*.

ignore-end-stop

Überspringe Qualitätsprüfung *Nicht übereinstimmender Punkt*.

ignore-end-question

Überspringe Qualitätsprüfung *Nicht übereinstimmendes Fragezeichen*.

ignore-end-semicolon

Überspringe Qualitätsprüfung *Nicht übereinstimmendes Semikolon*.

ignore-newline-count

Überspringe Qualitätsprüfung *Nicht übereinstimmende Zeilenumbrüche*.

ignore-plurals

Überspringe Qualitätsprüfung *Fehlende Pluralformen*.

ignore-placeholders

Überspringe Qualitätsprüfung *Platzhalter*.

ignore-punctuation-spacing

Überspringe Qualitätsprüfung *Satzzeichenabstand*.

ignore-regex

Überspringe Qualitätsprüfung *Regulärer Ausdruck*.

ignore-same-plurals

Überspringe Qualitätsprüfung *Identische Pluralformen*.

ignore-begin-newline

Überspringe Qualitätsprüfung *Zeilenumbruch am Anfang*.

ignore-begin-space

Überspringe Qualitätsprüfung *Leerzeichen am Anfang*.

ignore-end-newline

Überspringe Qualitätsprüfung *Zeilenumbruch am Ende*.

ignore-end-space

Überspringe Qualitätsprüfung *Leerzeichen am Ende*.

ignore-same

Überspringe Qualitätsprüfung *Unveränderte Übersetzung*.

ignore-safe-html

Überspringe Qualitätsprüfung *Unsicheres HTML*.

ignore-url

Überspringe Qualitätsprüfung *URL*.

ignore-xml-tags

Überspringe Qualitätsprüfung *XML-Auszeichnung*.

ignore-xml-invalid

Überspringe Qualitätsprüfung *XML-Syntax*.

ignore-zero-width-space

Überspringe Qualitätsprüfung *Leerzeichen ohne Breite*.

ignore-ellipsis

Überspringe Qualitätsprüfung *Auslassungspunkte*.

ignore-icu-message-format-syntax

Überspringe Qualitätsprüfung *ICU MessageFormat-Syntax*.

ignore-long-untranslated

Überspringe Qualitätsprüfung *Lange nicht übersetzt*.

ignore-multiple-failures

Überspringe Qualitätsprüfung *Mehrfach fehlgeschlagene Überprüfungen*.

ignore-unnamed-format

Überspringe Qualitätsprüfung *Mehrere unbenannte Variablen*.

ignore-optional-plural

Überspringe Qualitätsprüfung *Ohne Pluralformen*.

Bemerkung: Im Allgemeinen heißt die Regel `ignore-*` für jede Prüfung, wobei ihr Bezeichner verwendet wird, so dass Sie dies auch für Ihre benutzerdefinierten Prüfungen verwenden können.

Diese Markierungen werden sowohl in den *Komponentenkonfiguration*-Einstellungen, in den Einstellungen für die Ausgangszeichenkette als auch in der Übersetzungsdatei selbst (zum Beispiel in GNU gettext) verstanden.

2.12.3 Qualitätsprüfungen erzwingen

Neu in Version 3.11.

Sie können eine Liste von Prüfungen konfigurieren, die nicht ignoriert werden können, indem Sie *Erzwungene Qualitätsprüfungen* in *Komponentenkonfiguration* setzen. Jede aufgelistete Prüfung kann in der Benutzeroberfläche nicht abgelehnt werden und jede Zeichenkette, die diese Prüfung nicht besteht, wird als *Bearbeitung erforderlich* markiert (siehe *Übersetzungsstatus*).

Bemerkung: Das Einschalten der Qualitätsprüfung-Erzwingung aktiviert diese nicht automatisch. Die Prüfung kann durch Hinzufügen der entsprechenden Markierung zu den Zeichenketten- oder Komponenten-Markierungen aktiviert werden.

Siehe auch:

Zusätzliche Informationen über Ausgangszeichenketten, Übersetzungsmarkierungen

2.12.4 Schriftarten verwalten

Neu in Version 3.7.

Hinweis: In Weblate hochgeladene Schriften werden nur für die *Maximaler Umfang der Übersetzung*-Prüfung verwendet, sie haben keine Auswirkungen auf die Weblate-Benutzeroberfläche.

Die Qualitätsprüfung *Maximaler Umfang der Übersetzung*, die zur Berechnung der Größe des gerenderten Textes verwendet wird, benötigt eine Schriftart, die in Weblate geladen und mit einer Übersetzungsmarkierung ausgewählt wurde (siehe *Anpassen des Verhaltens mit Markierungen*).

Das Weblate-Schriftarten-Management-Tool in *Schriftarten* unter dem Menü *Verwalten* Ihres Übersetzungsprojekts bietet eine Schnittstelle zum Hochladen und Verwalten von Schriftarten. Sie können TrueType- oder OpenType-Schriften hochladen, Font-Gruppen einrichten und diese bei der Prüfung verwenden.

Die Schriftartgruppen ermöglichen es Ihnen, verschiedene Schriftarten für verschiedene Sprachen zu definieren, was normalerweise für nicht-lateinische Sprachen erforderlich ist:

Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

+

▾

...

WeblateOrg / Font groups / default-font

Font group

Name	default-font		
Default font	Source Sans 3 Bold		
Japanese	language override	Droid Sans Fallback Regular	Remove
Korean	language override	Droid Sans Fallback Regular	Remove
Delete			

Add language override

Language

----- ▾

Font

----- ▾

Save

Edit font group

Font group name

default-font

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

Source Sans 3 Bold ▾

Default font is used unless per language override matches.

Save

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Die Schriftartgruppen werden durch einen Namen identifiziert, der keine Leer- oder Sonderzeichen enthalten darf, damit er in der Prüfdefinition leicht verwendet werden kann:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Fonts

Font groups
 Fonts

Group name	Default font	Language overrides	
default-font	Source Sans 3 Bold	Japanese: Droid Sans Fallback Regular Korean: Droid Sans Fallback Regular	Edit

Add font group

Font group name

 Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

 Default font is used unless per language override matches.

Save

Powered by Weblate 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

Schriftfamilie und -stil werden nach dem Hochladen automatisch erkannt:

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Fonts / Droid Sans Fallback Regular

Font	
Font family	Droid Sans Fallback
Font style	Regular
File size	3939852
Created	now
Uploaded by	testuser
Used in groups	
Delete	

Powered by Weblate 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

In Weblate können Sie eine Reihe von Schriftarten laden:

Um die Schriftarten für die Überprüfung der Zeichenkettenlänge zu verwenden, übergeben Sie ihr die entsprechenden Markierungen (siehe [Anpassen des Verhaltens mit Markierungen](#)). Sie werden wahrscheinlich die folgenden benötigen:

max-length:N

Legt die maximale Breite in Pixeln fest.

font-weight:WEIGHT

Definiert die zu verwendende Schriftartgruppe durch Angabe ihres Bezeichners.

max-length:N

Definition der Schriftgröße in Pixeln.

2.12.5 Eigene Qualitätsprüfungen schreiben

Eine breite Palette von Qualitätsprüfungen ist eingebaut (siehe [Qualitätsprüfungen](#)), obwohl sie möglicherweise nicht alles abdecken, was Sie überprüfen wollen. Die Liste der durchgeführten Prüfungen kann mit `CHECK_LIST` angepasst werden, und Sie können auch eigene Prüfungen hinzufügen.

1. Unterklasse von `weblate.checks.Check` erstellen
2. Legen Sie ein paar Attribute fest.
3. Implementieren Sie entweder die Methode `check` (wenn Sie in Ihrem Code mit Pluralen umgehen wollen) oder die Methode `check_single` (die dies für Sie erledigt).

Einige Beispiele:

Um benutzerdefinierte Prüfungen zu installieren, geben Sie in `CHECK_LIST` einen vollqualifizierten Pfad zur Python-Klasse an, siehe [Benutzerdefinierte Qualitätsprüfungen](#), [Erweiterungen](#) und [automatische Korrekturen](#).

Prüfen, ob der Übersetzungstext „foo“ nicht enthält

Es handelt sich um eine recht einfache Prüfung, bei der lediglich überprüft wird, ob in der Übersetzung die Zeichenfolge „foo“ fehlt.

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Simple quality check example."""

from django.utils.translation import gettext_lazy as _
from weblate.checks.base import TargetCheck

class FooCheck(TargetCheck):
    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_single(self, source, target, unit):
        return "foo" in target
```

Prüfen, ob sich die Pluralformen des tschechischen Übersetzungstextes unterscheiden

Prüfen Sie anhand der Sprachinfo, ob die beiden Pluralformen im Tschechischen nicht gleich sind.

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Quality check example for Czech plurals."""

from django.utils.translation import gettext_lazy as _
from weblate.checks.base import TargetCheck

class PluralCzechCheck(TargetCheck):
    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_target_unit(self, sources, targets, unit):
        if self.is_language(unit, ("cs",)):
            return "foo" in targets
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)


```
        return targets[1] == targets[2]
    return False


def check_single(self, source, target, unit):
    """We don't check target strings here."""
    return False
```

2.13 Automatische Vorschläge konfigurieren

Geändert in Version 4.13: Prior to Weblate 4.13, the services were configured in the *Konfiguration*.

Die Unterstützung für verschiedene maschinelle Übersetzungen und Übersetzungsspeicher-Dienste ist integriert. Jeder Dienst kann vom Administrator für die gesamte Website oder in den Projekteinstellungen aktiviert werden:


Dashboard
Projects ▾
Languages ▾
Checks ▾
🔍
+
🌐
⋮


WeblateOrg / Automatic suggestions

Configured automatic suggestion services ⓘ

There are no services currently installed.

Available automatic suggestion services ⓘ

AWS ⓘ	Install
Amagama ⓘ	Install
Apertium APy ⓘ	Install
Baidu ⓘ	Install
DeepL ⓘ	Install
Glosbe ⓘ	Install
Google Translate ⓘ	Install
Google Translate API v3 ⓘ	Install
IBM ⓘ	Install
LibreTranslate ⓘ	Install
Microsoft Terminology ⓘ	Install
Microsoft Translator ⓘ	Install
ModernMT ⓘ	Install
MyMemory ⓘ	Install
Netease Sight ⓘ	Install
SAP Translation Hub ⓘ	Install
Weblate ⓘ	Install
Weblate Translation Memory ⓘ	Install
Yandex ⓘ	Install
Youdao Zhiyun ⓘ	Install
tmserver ⓘ	Install

Some services will ask for additional configuration during installation.

Bemerkung: They come subject to their terms of use, so ensure you are allowed to use them how you want.

Die Dienste übersetzen aus der Ausgangssprache, die unter *Komponentenkonfiguration* konfiguriert ist, siehe *Ausgangssprache*.

Siehe auch:

Automatische Vorschläge

2.13.1 Amagama

Dienst-ID

amagama

Konfiguration

Dieser Dienst hat keine Konfiguration.

Special installation of *tmserver* run by the authors of Virtaal.

Siehe auch:

Installing amaGama, Amagama, amaGama-Übersetzungsspeicher

2.13.2 Apertium APy

Dienst-ID

apertium-apy

Konfiguration

url	API-URL
-----	---------

Eine Libre-Software-Plattform für maschinelle Übersetzung, die Übersetzungen in eine begrenzte Anzahl von Sprachen liefert.

Der empfohlene Weg, Apertium zu nutzen, ist der Betrieb eines eigenen Apertium-APy-Servers.

Siehe auch:

Apertium-Website, *Apertium APy-Dokumentation*

2.13.3 AWS

Neu in Version 3.1.

Dienst-ID

aws

Konfiguration

key	Zugangsschlüssel-ID
secret	Geheimer API-Schlüssel
region	Name der Region

Amazon Translate ist ein neuronaler maschineller Übersetzungsdienst für die Übersetzung von Texten aus dem und ins Englische in einer Vielzahl von unterstützten Sprachen.

Siehe auch:[Amazon Translate Dokumentation](#)

2.13.4 Baidu

Neu in Version 3.2.

Dienst-ID

baidu

Konfiguration

key	Client-ID
secret	Client-Geheimnis

Maschineller Übersetzungsdienst von Baidu.

Dieser Dienst verwendet eine API und Sie müssen eine ID und einen API-Schlüssel von Baidu erhalten, um ihn zu nutzen.

Siehe auch:[Baidu Translate API](#)

2.13.5 DeepL

Neu in Version 2.20.

Dienst-ID

deepl

Konfiguration

url	API-URL
key	API-Schlüssel

DeepL ist ein kostenpflichtiger Dienst, der eine gute maschinelle Übersetzung für einige Sprachen bietet. Sie müssen ein *DeepL API*-Abonnement erwerben oder Sie können den alten *DeepL Pro (classic)*-Plan verwenden.

API-URL zur Verwendung mit dem DeepL-Dienst. Zum Zeitpunkt des Schreibens gibt es die v1-API sowie eine kostenlose und eine kostenpflichtige Version der v2-API.

<https://api.deepl.com/v2/> (Voreinstellung in Weblate)

Ist für die API-Nutzung des kostenpflichtigen Angebots gedacht, und das Abonnement ist nutzungsabhängig.

<https://api-free.deepl.com/v2/>

Ist für die API-Nutzung des kostenlosen Angebots gedacht, und das Abonnement ist nutzungsabhängig.

<https://api.deepl.com/v1/>

Ist für CAT-Tools gedacht und kann mit einem Benutzerabonnement verwendet werden.

Früher wurde Weblate von DeepL als CAT-Tool eingestuft und sollte daher die v1-API verwenden, aber jetzt soll es die v2-API verwenden. Daher ist v2 voreingestellt, und Sie können es auf v1 ändern, falls Sie ein bestehendes CAT-Abonnement haben und möchten, dass Weblate dieses nutzt.

Am einfachsten finden Sie dies heraus, indem Sie eine URL wie die folgende in Ihrem Browser öffnen:

https://api.deepl.com/v2/translate?text=Hello&target_lang=FR&auth_key=XXX

Ersetzen Sie die XXX durch Ihren auth_key. Wenn Sie ein JSON-Objekt erhalten, das „Bonjour“ enthält, haben Sie die richtige URL; wenn nicht, probieren Sie die anderen drei aus.

Weblate unterstützt DeepL-Formalität, es wählt eine passende basierend auf der Sprache (zum Beispiel gibt es `de@formal` und `de@informal`).

Siehe auch:

[DeepL-Website](#), [DeepL-Preise](#), [DeepL-API-Dokumentation](#)

2.13.6 Glosbe

Dienst-ID

`glosbe`

Konfiguration

Dieser Dienst hat keine Konfiguration.

Kostenloses Wörterbuch und Übersetzungsspeicher für fast alle lebenden Sprachen.

Die Nutzung der API ist kostenlos, aber die Verwendung der Übersetzungen unterliegt der Lizenz der verwendeten Datenquelle. Es gibt eine Begrenzung der Aufrufe, die von einer IP in einem bestimmten Zeitraum durchgeführt werden können, um Missbrauch zu verhindern.

Siehe auch:

[Glosbe-Website](#)

2.13.7 Google Translate

Dienst-ID

`google-translate`

Konfiguration

<code>key</code>	API-Schlüssel
------------------	---------------

Maschineller Übersetzungsdienst von Google.

Dieser Dienst verwendet die Google Translation API, und Sie müssen einen API-Schlüssel erhalten und die Rechnungsstellung in der Google-API-Konsole aktivieren.

Siehe auch:

[Google Translate-Dokumentation](#)

2.13.8 Google Translate API v3

Dienst-ID

`google-translate-api-v3`

Konfiguration

<code>credentials</code>	Google Translate Kontoinformationen
<code>project</code>	Google Translate Projekt
<code>location</code>	Google Translate Gebiet

Maschineller Übersetzungsdienst, der von Google Cloud-Diensten bereitgestellt wird.

Siehe auch:

[Google translate documentation](#), [Authenticate to Cloud services using client libraries](#), [Creating Google Translate project](#), [Google Cloud App Engine locations](#)

2.13.9 LibreTranslate

Neu in Version 4.7.1.

Dienst-ID

`libretranslate`

Konfiguration

<code>url</code>	API-URL
<code>key</code>	API-Schlüssel

LibreTranslate ist ein freier und quelloffener Dienst für maschinelle Übersetzungen. Die öffentliche Instanz erfordert einen API-Schlüssel, aber LibreTranslate kann selbst gehostet werden, und es gibt mehrere Mirrors, welche die API kostenlos nutzen.

<https://libretranslate.com/> (offizielle öffentliche Instanz)

Für die Nutzung außerhalb der Website ist ein API-Schlüssel erforderlich.

Siehe auch:

[LibreTranslate-Website](#), [LibreTranslate-Repository](#), [LibreTranslate-Mirrors](#)

2.13.10 Microsoft Terminology

Neu in Version 2.19.

Dienst-ID

`microsoft-terminology`

Konfiguration

Dieser Dienst hat keine Konfiguration.

Die Microsoft-Terminologiedienst-API ermöglicht Ihnen den programmgesteuerten Zugriff auf die im Sprachportal verfügbare Terminologie, Definitionen und Benutzeroberflächen-Zeichenketten über einen Webdienst.

Siehe auch:

[Microsoft-Terminologiedienst-API](#)

2.13.11 Microsoft Translator

Neu in Version 2.10.

Dienst-ID

`microsoft-translator`

Konfiguration

key	API-Schlüssel	
base_url	Basis-URL der Anwendung	Verfügbare Auswahlmöglichkeiten: <i>api.cognitive.microsofttranslator.com</i> – Global (nicht regional) <i>api-apc.cognitive.microsofttranslator.com</i> – Asien-Pazifik <i>api-eur.cognitive.microsofttranslator.com</i> – Europa <i>api-nam.cognitive.microsofttranslator.com</i> – Nordamerika <i>api.translator.azure.cn</i> – China <i>api.cognitive.microsofttranslator.us</i> – Azure US-Regierungscloud
endpoint	Authentifizierung URL	Regional oder Multi-Service kann über das unten stehende Feld Region angegeben werden. Verfügbare Auswahlmöglichkeiten: <i>api.cognitive.microsoft.com</i> – Global <i>api.cognitive.azure.cn</i> – China <i>api.cognitive.microsoft.us</i> – Azure US-Regierungscloud
region	Authentifizierung Region	

Maschineller Übersetzungsdienst, der von Microsoft im Azure-Portal als einer der Cognitive-Services bereitgestellt wird.

Weblate implementiert Translator API V3.

Translator Text API V2

Der Schlüssel, den Sie mit Translator-API V2 verwenden, kann auch mit API 3 verwendet werden.

Translator Text API V3

Sie müssen sich im Azure-Portal registrieren und den Schlüssel verwenden, den Sie dort erhalten. Bei neuen Azure-Schlüsseln müssen Sie auch `region` auf das Gebietsschema Ihres Dienstes einstellen.

Hinweis: Für Azure China verwenden Sie bitte Ihren Endpunkt aus dem Azure-Portal.

Siehe auch:

Cognitive-Services – Text-Translation-API, Microsoft-Azure-Portal, Basis-URLs, „Authentifizierung mit einer Multi-Service-Ressource“ ` „Authentifizierung mit einem Zugriffstoken“ Abschnitt <https://learn.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-reference#authenticating-with-an-access-token>

2.13.12 ModernMT

Neu in Version 4.2.

Dienst-ID

modernmt

Konfiguration

url	API-URL
key	API-Schlüssel

Siehe auch:[ModernMT-API](#),

2.13.13 MyMemory

Dienst-ID`mymemory`**Konfiguration**

<code>email</code>	Kontakt-E-Mail
<code>username</code>	Benutzername
<code>key</code>	API-Schlüssel

Großer Übersetzungsspeicher mit maschineller Übersetzung.

Die kostenlose, anonyme Nutzung ist derzeit auf 100 Anfragen/Tag oder auf 1000 Anfragen/Tag begrenzt, wenn Sie eine Kontakt-E-Mail-Adresse in `email` angeben. Sie können auch mehr Anfragen stellen.

Siehe auch:[MyMemory-Website](#)

2.13.14 Netease Sight

Neu in Version 3.3.

Dienst-ID`netease-sight`**Konfiguration**

<code>key</code>	Client-ID
<code>secret</code>	Client-Geheimnis

Maschineller Übersetzungsdienst von NetEase.

Dieser Dienst verwendet eine API, und Sie müssen Schlüssel und Geheimnis von NetEase abrufen.

Siehe auch:[NetEase Sight Übersetzungsplattform](#)

2.13.15 SAP Translation Hub

Dienst-ID`sap-translation-hub`**Konfiguration**

url	API-URL	
key	API-Schlüssel	
user-	SAP-	
na-	Benutzername	
me		
pass-	SAP-Passwort	
word		
enabl	Maschinelle	
	Übersetzung	
	aktivieren	
do-	Übersetzungs-	Die ID einer Übersetzungsdomain, zum Beispiel BC. Wenn Sie kei-
main	domain	ne Domain angeben, sucht die Methode nach Übersetzungen in allen verfügbaren Domains.

Maschineller Übersetzungsdienst von SAP.

Sie müssen über ein SAP-Benutzerkonto verfügen (und den SAP Translation Hub in der SAP Cloud Platform aktiviert haben), um diesen Dienst nutzen zu können.

Sie können auch festlegen, ob neben der Begriffsdatenbank auch maschinelle Übersetzungsdienste verwendet werden sollen.

Bemerkung: To access the Sandbox API, you need to set `url` and `key`.

To access the productive API, you need to set `url`, `username` and `password`.

Siehe auch:

[SAP Translation Hub API](#), [Building the Base URL of SAP Translation Hub](#)

2.13.16 tmserver

Dienst-ID

tmserver

Konfiguration

url	API-URL
-----	---------

You can run your own translation memory server by using the one bundled with Translate-toolkit and let Weblate talk to it. You can also use it with an amaGama server, which is an enhanced version of tmserver.

1. Zuerst sollten Sie einige Daten in den Übersetzungsspeicher importieren:

```
build_tmdb -d /var/lib/tm/db -s en -t cs locale/cs/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t de locale/de/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t fr locale/fr/LC_MESSAGES/django.po
```

2. Starten Sie tmserver, um Ihre Anfragen abzuhören:

```
tmserver -d /var/lib/tm/db
```

3. Configure Weblate to talk to it, the default URL is `http://localhost:8888/tmserver/`.

Siehe auch:

[tmserver](#) [Installing amaGama](#), [Amagama](#), [Amagama Translation Memory](#)

2.13.17 IBM Watson Language Translator

Dienst-ID

ibm

Konfiguration

url	API-URL
key	API-Schlüssel

IBM Watson Language Translator translates text from one language to another. The service offers multiple domain-specific models.

Siehe auch:

Watson Language Translator, IBM Cloud API Docs

2.13.18 Weblate

Dienst-ID

weblate

Konfiguration

Dieser Dienst hat keine Konfiguration.

Der maschinelle Übersetzungsdienst von Weblate kann Übersetzungen für Zeichenketten liefern, die bereits in Weblate übersetzt wurden. Es wird nach exakten Übereinstimmungen in den vorhandenen Zeichenketten gesucht.

2.13.19 Weblate Translation Memory

Neu in Version 2.20.

Dienst-ID

weblate-translation-memory

Konfiguration

Dieser Dienst hat keine Konfiguration.

Verwenden Sie *Übersetzungsspeicher* als maschinellen Übersetzungsdienst. Jede Zeichenkette, die in der Vergangenheit übersetzt (oder in den Übersetzungsspeicher hochgeladen) wurde, kann auf diese Weise übersetzt werden.

2.13.20 Yandex

Dienst-ID

yandex

Konfiguration

key	API-Schlüssel
-----	---------------

Maschineller Übersetzungsdienst von Yandex.

This service uses a Translation API, and you need to obtain an API key from Yandex.

Siehe auch:

Yandex Translate API, Powered by Yandex.Translate

2.13.21 Youdao Zhiyun

Neu in Version 3.2.

Dienst-ID

youdao-zhiyun

Konfiguration

key	Client-ID
secret	Client-Geheimnis

Maschineller Übersetzungsdienst von Youdao.

This service uses an API, and you need to obtain an ID and an API key from Youdao.

Siehe auch:

[Youdao Zhiyun Natural Language Translation Service](#)

2.13.22 Benutzerdefinierte maschinelle Übersetzung

Sie können auch Ihre eigenen maschinellen Übersetzungsdienste mit ein paar Zeilen Python-Code implementieren. Dieses Beispiel implementiert maschinelle Übersetzung in einer festen Liste von Sprachen mit dem Python-Modul `dictionary`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation

class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

    def download_languages(self):
        """Return list of languages your machine translation supports."""
        return {"cs"}

    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```


Sie können Ihre eigene Klasse in `WEBLATE_MACHINERY` auflisten und Weblate wird diese dann verwenden.


2.14 Erweiterungen

Neu in Version 2.19.

Erweiterungen bieten Möglichkeiten, den Ablauf der Übersetzung anzupassen und zu automatisieren. Administratoren können Erweiterungen über das Menü *Verwaltung* ↓ *Erweiterungen* der jeweiligen Übersetzungskomponente hinzufügen und verwalten.

Hinweis: Sie können Erweiterungen auch mit *API*, *DEFAULT_ADDONS*, oder *install_addon* konfigurieren.



[Weblate](#)
[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)


[WeblateOrg](#) / [Language names](#) / [Add-ons](#)

Installed add-ons


There are no add-ons currently installed.

Available add-ons


Automatic translation

Automatically translates strings using machine translation or other components.


Install


Add missing languages

Ensures a consistent set of languages is used for all components within a project.

project wide


Install


Component discovery

Automatically adds or removes project components based on file changes in the version control system.


repository wide

Install


Bulk edit


Bulkedit flags, labels, or states of strings.

Install


Statistics generator


Generates a file containing detailed info about the translation status.

Install


Prefill translation with source


Fills in translation strings with source string.

Install


Pseudolocale generation


Generates a translation by adding prefix and suffix to source strings automatically.

Install


Contributors in comment


Updates the comment part of the PO file header to include contributor names and years of contributions.

Install


Customize gettext output


Allows customization of gettext output behavior, for example line wrapping.

Install


Generate MO files


Automatically generates a MO file for every changed PO file.

Install


Update PO files to match POT (msgmerge)

Updates all PO files (as configured by "File mask") to match the POT file (as configured by "Template for new translations") using msgmerge.


Install


Squash Git commits

Squash Git commits prior to pushing changes.

repository wide


Install


Stale comment removal

Set a timeframe for removal of comments.

project wide

Install


Stale suggestion removal

Set a timeframe for removal of suggestions.

project wide

Install

Some add-ons will ask for additional configuration during installation.

2.14.1 Integrierte Erweiterungen

Automatische Übersetzung

Neu in Version 3.9.

Erweiterungs-ID

`weblate.autotranslate.autotranslate`

Konfiguration

<code>mode</code>	Automatischer Übersetzungsmodus	Verfügbare Auswahlmöglichkeiten: <code>suggest</code> – Als Vorschlag hinzufügen <code>translate</code> – Als Übersetzung hinzufügen <code>fuzzy</code> – Als „Zur Bearbeitung“ hinzufügen
<code>filter</code>	Suchfilter	Bitte beachten Sie, dass bei der Übersetzung aller Zeichenketten alle vorhandenen Übersetzungen verworfen werden. Verfügbare Auswahlmöglichkeiten: <code>a`all`</code> – Alle Zeichenketten <code>nottranslated</code> – Nicht übersetzte Zeichenketten <code>todo</code> – Unfertige Zeichenketten <code>fuzzy</code> – Zur Bearbeitung markierte Zeichenketten <code>check:inconsistent</code> – Prüfung fehlgeschlagen: Inkonsistent
<code>auto_source</code>	Quelle der automatisierten Übersetzungen	Verfügbare Auswahlmöglichkeiten: <code>others</code> – Andere Übersetzungskomponenten <code>mt</code> – Maschinelle Übersetzung
<code>component</code>	Komponenten	Geben Sie den Slug einer Komponente ein, die als Ausgangszeichenkette verwendet werden soll. Lassen Sie das Feld leer, um alle Komponenten im aktuellen Projekt zu verwenden.
<code>enable</code>	Vorschläge aus automatischer Übersetzung	
<code>threshold</code>	Score-schwellwert	

Trigger

Komponenten-Update, täglich

Übersetzt automatisch Zeichenketten mithilfe maschineller Übersetzung oder Übersetzungen aus anderen Komponenten.

Es wird ausgelöst:

- Wenn neue Zeichenketten in einer Komponente erscheinen.
- Einmal monatlich für jede Komponente, dies kann mit `BACKGROUND_TASKS` konfiguriert werden.

Siehe auch:

Automatische Übersetzung, Übersetzungen aller Komponenten gleich halten

JavaScript-Lokalisierung CDN

Neu in Version 4.2.

Erweiterungs-ID

`weblate.cdn.cdnjs`

Konfiguration

th-res-hold	Übersetzungsgrenzwert	Schwellenwert für die Einbindung von Übersetzungen.
css_selector	CSS-Selektor	CSS-Selektor zur Erkennung lokalisierbarer Elemente.
cookie_name	Name des Sprachcookies	Name des Cookies, der die Sprachpräferenz speichert.
files	Zeichenketten aus HTML-Dateien extrahieren	Liste der Dateinamen im aktuellen Repository oder Remote-URLs, die nach übersetzbaren Zeichenketten analysiert werden sollen.

Trigger

daily, repository post-commit, repository post-update

Veröffentlicht Übersetzungen in einem Content Delivery-Netzwerk zur Verwendung in der JavaScript- oder HTML-Lokalisierung.

Kann verwendet werden, um statische HTML-Seiten zu übersetzen oder um die Übersetzung in den JavaScript-Code zu laden.

Erzeugt eine eindeutige URL für Ihre Komponente, die Sie in HTML-Seiten einfügen können, um diese zu lokalisieren. Siehe `weblate-cdn` für weitere Details.

Siehe auch:

`cdn-addon-config`, `weblate-cdn`, `cdn-addon-extract`, `cdn-addon-html`

Leere Zeichenketten entfernen

Neu in Version 4.4.

Erweiterungs-ID

`weblate.cleanup.blank`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

repository post-commit, repository post-update

Entfernt Zeichenketten ohne Übersetzung aus den Übersetzungsdateien.

Verwenden Sie dies, um leere Zeichenketten in Übersetzungsdateien zu vermeiden (z. B. wenn Ihre Lokalisierungsbibliothek sie als fehlend anzeigt, anstatt auf die Ausgangszeichenkette zurückzugreifen).

Siehe auch:

Aktualisiert Weblate neben den Übersetzungen auch die Übersetzungsdateien?

Übersetzungsdateien bereinigen

Erweiterungs-ID

`weblate.cleanup.generic`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

repository pre-commit, repository post-update

Aktualisieren Sie alle Übersetzungsdateien so, dass sie mit der einsprachigen Basisdatei übereinstimmen. Für die meisten Dateiformate bedeutet dies, veraltete Übersetzungsschlüssel, die in der Basisdatei nicht mehr vorhanden sind, zu entfernen.

Siehe auch:

Aktualisiert Weblate neben den Übersetzungen auch die Übersetzungsdateien?

Fehlende Sprachen hinzufügen

Erweiterungs-ID

`weblate.consistency.languages`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

daily, repository post-add

Stellt sicher, dass ein konsistenter Satz von Sprachen für alle Komponenten innerhalb eines Projekts verwendet wird.

Fehlende Sprachen werden alle 24 Stunden überprüft, und wenn neue Sprachen in Weblate hinzugefügt werden.

Im Gegensatz zu den meisten anderen wirkt sich diese Erweiterung auf das gesamte Projekt aus.

Hinweis: Übersetzen Sie automatisch die neu hinzugefügten Zeichenketten mit *Automatische Übersetzung*.

Komponentenerkennung

Erweiterungs-ID

weblate.discovery.discovery

Konfiguration

match	Regulärer Ausdruck zum Abgleich von Übersetzungsdateien	
file_format	Dateiformat	
name_temp	Anpassen des Komponentennamens	
base_file	Definition des einsprachigen Basisdateinamens	Für zweisprachige Übersetzungsdateien leer lassen.
new_base	Definition der Basisdatei für neue Übersetzungen	Dateiname, der zum Anlegen neuer Übersetzungen verwendet wird. Für gettext .pot-Datei wählen.
intermediate_temp	Zwischensprache-datei	Dateiname der Zwischenübersetzungsdatei. In den meisten Fällen handelt es sich um eine von Entwicklern bereitgestellte Übersetzungsdatei, die beim Erstellen der eigentlichen Ausgangszeichenketten verwendet wird.
language_remove_add	Sprachen-Filter Erweiterungen der Hauptkomponente zu den neu erstellten klonen	Regulärer Ausdruck, gegen den die Übersetzungsdateien beim Scannen nach Dateimaske gefiltert wird.
remove	Komponenten für nicht vorhandene Dateien entfernen	
confirm	Ich bestätige, dass die obigen Übereinstimmungen korrekt aussehen	

Trigger

Repository nach dem Update

Automatisches Hinzufügen oder Entfernen von Projektkomponenten basierend auf den Dateiänderungen im Versionskontrollsystem.

Wird jedes mal ausgelöst, wenn das VCS aktualisiert wird, und ähnelt ansonsten dem Verwaltungsbefehl `import_project`. Auf diese Weise können Sie mehrere Übersetzungskomponenten innerhalb eines VCS verfolgen.

Der Abgleich erfolgt über reguläre Ausdrücke, die eine komplexe Konfiguration ermöglichen, für die jedoch einige Kenntnisse erforderlich sind. Einige Beispiele für häufige Anwendungsfälle finden Sie im Hilfebereich der Erweiterung.

Sobald Sie auf *Speichern* klicken, wird eine Vorschau der passenden Komponenten angezeigt, anhand derer Sie überprüfen können, ob die Konfiguration tatsächlich Ihren Anforderungen entspricht:

Weblate

Dashboard

Projects

Languages

Checks

+

+

+

+

🌐 WeblateOrg

/

Language names

/

Add-ons

/

Component discovery

Configure add-on

ⓘ

Please review and confirm the matched components.

Component

Matched files

The following components would be created

Djangojs (**djangojs**)

File mask: `weblate/locale/*/LC_MESSAGES/djangojs.po`
`weblate/locale/cs/LC_MESSAGES/djangojs.po` (cs)
`weblate/locale/hu/LC_MESSAGES/djangojs.po` (hu)
`weblate/locale/he/LC_MESSAGES/djangojs.po` (he)

Django (**django**)

File mask: `weblate/locale/*/LC_MESSAGES/django.po`
`weblate/locale/hu/LC_MESSAGES/django.po` (hu)
`weblate/locale/cs/LC_MESSAGES/django.po` (cs)
`weblate/locale/he/LC_MESSAGES/django.po` (he)

☐ I confirm the above matches look correct

Regular expression to match translation files against

`weblate/locale/(?P<language>["/"]*)/LC_MESSAGES/(?P<component>["/"]*)\.po`

File format

gettext PO file

Customize the component name

`{{ component|title }}`

Define the monolingual base filename

Leave empty for bilingual translation files.

Define the base file for new translations

`weblate/locale/{{ component }}.pot`

Filename of file used for creating new translations. For gettext choose .pot file.

Intermediate language file

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Language filter

`^(cs|he|hu)$`

Regular expression to filter translation files against when scanning for file mask.

☒ Clone add-ons from the main component to the newly created ones

☐ Remove components for inexistent files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description
<code>(?P<language>["/"].)*/(?P<component>["/"]*)\.po</code>	<code>cs/application.po</code> <code>cs/website.po</code> <code>de/application.po</code> <code>de/website.po</code>	One folder per language containing translation files for components.
<code>locale/(?P<language>["/"].)*LC_MESSAGES/(?P<component>["/"]*)\.po</code>	<code>locale/cs/LC_MESSAGES/application.po</code> <code>locale/cs/LC_MESSAGES/website.po</code> <code>locale/de/LC_MESSAGES/application.po</code> <code>locale/de/LC_MESSAGES/website.po</code>	Usual structure for storing gettext PO files.
<code>src/locale/(?P<component>["/"]*)\.(?P<language>["/"].)*\.po</code>	<code>src/locale/application.cs.po</code> <code>src/locale/website.cs.po</code> <code>src/locale/application.de.po</code> <code>src/locale/website.de.po</code>	Using both component and language name within filename.
<code>locale/(?P<language>["/"].)*/(?P<component>["/"]*)/(?P<language>["/"].)*\.po</code>	<code>locale/cs/application/cs.po</code> <code>locale/cs/website/cs.po</code> <code>locale/de/application/de.po</code> <code>locale/de/website/de.po</code>	Using language in both path and filename.
<code>res/values-(?P<language>["/"].)*strings-(?P<component>["/"]*)\.xml</code>	<code>res/values-cs/strings-about.xml</code> <code>res/values-cs/strings-help.xml</code> <code>res/values-de/strings-about.xml</code> <code>res/values-de/strings-help.xml</code>	Android resource strings, split into several files.
<code>(?P<originalHierarchy>.*)(?P<component>["/"]*)src/main/resources/ApplicationResources_(?P<language>["/"].)*.properties</code>	<code>parent/module1/submodule/src/main/resources/ApplicationResources_fr.properties</code> <code>parent/module1/submodule/src/main/resources/ApplicationResource_es.properties</code> <code>parent/module2/src/main/resources/ApplicationResource_de.properties</code> <code>parent/module2/src/main/resources/ApplicationResource_ro.properties</code>	Multi-module Maven project with Java properties translations.

You can use Django template markup in both component name and the monolingual base filename, for example:

`{{ component }}`
Component filename match
`{{ component|title }}`
Component filename with upper case first letter

Save

Powered by Weblate 4.16

About Weblate

Legal

Contact

Documentation

Donate to Weblate

2.14. Erweiterungen

335

Hinweis: Die Erweiterung zur Komponentenerkennung verwendet *Weblate-interne URLs*. Es ist ein bequemer Weg, um das VCS-Setup zwischen mehreren Komponenten zu teilen. Verknüpfte Komponenten verwenden das lokale Repository der Hauptkomponente, das durch Eintragen von `weblate://project/main-component` in das Feld *Quellcode-Repository* (in *Verwalten* ↓ *Einstellungen* ↓ *Versionsverwaltung*) der jeweiligen Komponente eingerichtet wird. Dies spart Zeit bei der Konfiguration und auch Systemressourcen.

Siehe auch:

Template markup

Massenbearbeitung

Neu in Version 3.11.

Erweiterungs-ID

`weblate.flags.bulk`

Konfiguration

q	Abfrage	
state	Zu setzender Status	Verfügbare Auswahlmöglichkeiten: -1 – Nicht ändern 10 – Bearbeitung erforderlich 20 – Übersetzt 30 – Genehmigt
add_flag	Übersetzungs- markierung zum Hinzufügen	
remove_flag	Übersetzungs- markierung zum Entfernen	
add_label	Labels zum Hinzufügen	
remove_label	Labels zum Entfernen	

Trigger

Komponentenaktualisierung

Massenbearbeitung von Markierungen, Labels oder den Zeichenkettenstatus.

Automatisieren Sie die Beschriftung, indem Sie mit der Suchanfrage `NOT has:label` beginnen und so lange Beschriftungen hinzufügen, bis alle Zeichenketten alle erforderlichen Beschriftungen aufweisen. Andere automatisierte Vorgänge für Weblate-Metadaten können ebenfalls durchgeführt werden.

Beispiele:

Tab. 5: Neue Zeichenketten automatisch beschriften

Suchanfrage	<code>NOT has:label</code>
Labels zum Hinzufügen	<code>recent</code>

Tab. 6: Markierung aller App-Store-Metadatendateien
Änderungsprotokoll-Zeichenketten als schreibgeschützt

Suchanfrage	<code>language:en AND key:changelogs/</code>
Übersetzungsmarkierung zum Hinzufügen	<code>read-only</code>

Siehe auch:

Massenbearbeitung, Anpassen des Verhaltens mit Markierungen, labels

Unveränderte Übersetzungen als „bearbeitungsbedürftig“ markieren

Neu in Version 3.1.

Erweiterungs-ID

`weblate.flags.same_edit`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

unit post-create

Jedes Mal wenn eine neue übersetzbare Zeichenkette aus dem VCS importiert wird und mit der Ausgangszeichenkette übereinstimmt, wird sie in Weblate als zu bearbeiten markiert. Dies ist besonders nützlich für Dateiformate, die nicht übersetzte Ausgangszeichenketten enthalten.

Hinweis: Sie können auch die *Unveränderte Übersetzung* Prüfung verschärfen, indem Sie `strict-same` zu *Übersetzungsmarkierungen* hinzufügen.

Siehe auch:

Übersetzungsstatus

Neue Ausgangszeichenketten als „bearbeitungsbedürftig“ markieren**Erweiterungs-ID**

`weblate.flags.source_edit`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

unit post-create

Bei jedem Import einer neuen Ausgangszeichenkette aus dem VCS wird diese in Weblate als zu bearbeiten gekennzeichnet. Auf diese Weise können Sie die von den Entwicklern geschriebenen Ausgangszeichenketten einfach filtern und bearbeiten.

Siehe auch:

Übersetzungsstatus

Neue Übersetzungen als „bearbeitungsbedürftig“ markieren**Erweiterungs-ID**

`weblate.flags.target_edit`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

unit post-create

Bei jedem Import einer neuen übersetzbaren Zeichenkette aus dem VCS wird diese in Weblate als zu bearbeiten gekennzeichnet. Auf diese Weise können Sie die von den Entwicklern erstellten Übersetzungen einfach filtern und bearbeiten.

Siehe auch:

[Übersetzungsstatus](#)

Statistikgenerator

Erweiterungs-ID

`weblate.generate.generate`

Konfiguration

<code>filename</code>	Name der erzeugten Datei
<code>template</code>	Inhalt der erzeugten Datei

Trigger

Repository Pre-Commit

Erzeugt eine Datei mit detaillierten Informationen zum Übersetzungsstatus.

Sie können eine Django-Vorlage sowohl im Dateinamen als auch im Inhalt verwenden, siehe [Template markup](#) für eine detaillierte Beschreibung des Markups.

Zum Beispiel das Erzeugen einer Zusammenfassungsdatei für jede Übersetzung:

Name der erzeugten Datei

`locale/{{ language_code }}.json`

Inhalt

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}"
}
```

Siehe auch:

[Template markup](#)

Übersetzung mit Quelle vorbelegen

Neu in Version 4.11.

Erweiterungs-ID

`weblate.generate.prefill`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

Komponenten-Update, täglich

Füllt Übersetzungszeichenketten mit der Ausgangszeichenkette aus.

Alle nicht übersetzten Zeichenketten in der Komponente werden mit der Ausgangszeichenkette gefüllt und als zu bearbeiten markiert. Verwenden Sie dies, wenn in den Übersetzungsdateien keine leeren Zeichenketten enthalten sein dürfen.

Erzeugung von Pseudolokalen

Neu in Version 4.5.

Erweiterungs-ID

`weblate.generate.pseudolocale`

Konfiguration

<code>source</code>	Ausgangszeichenketten	
<code>target</code>	Zielübersetzung	Alle Zeichenketten in dieser Übersetzung werden überschrieben
<code>prefix</code>	Festes Zeichenkettenpräfix	
<code>var_prefix</code>	Variables Zeichenkettenpräfix	
<code>suffix</code>	Festes Zeichenkettensuffix	
<code>var_suffix</code>	Variables Zeichenkettensuffix	
<code>var_multiplier</code>	Variabler Teilemultiplikator	Wie oft der variable Teil wiederholt werden soll, hängt von der Länge der Ausgangszeichenkette ab.
<code>include_real</code>	Schreibgeschützte Zeichenketten einschließen	

Trigger

Komponenten-Update, täglich

Erzeugt eine Übersetzung durch automatisches Hinzufügen von Präfix und Suffix zu Ausgangszeichenketten.

Pseudolokalisierungen sind nützlich, um Zeichenketten zu finden, die nicht für die Lokalisierung vorbereitet sind. Dazu werden alle übersetzbaren Ausgangszeichenketten geändert, so dass es einfach ist, unveränderte Zeichenketten zu erkennen, wenn die Anwendung in der Pseudolokalsprache ausgeführt wird.

Es ist auch möglich, Zeichenketten zu finden, deren lokalisierte Entsprechungen nicht in das Layout passen.

Die Verwendung der variablen Teile ermöglicht es, nach Zeichenketten zu suchen, die nach der Lokalisierung möglicherweise nicht in die Benutzeroberfläche passen – sie erweitert den Text auf der Grundlage der Länge der Ausgangszeichenkette. Die variablen Teile werden um die Länge des Textes multipliziert mit dem Multiplikator wiederholt. Zum Beispiel wird `Hello world` mit dem variablen Suffix `_` und dem variablen Multiplikator von 1 zu `Hello world_____` – das Suffix wird für jedes Zeichen in der Ausgangszeichenkette einmal wiederholt.

Die Zeichenketten werden nach folgendem Muster erzeugt:

Festes Zeichenkettenpräfix Variables Zeichenkettenpräfix Ausgangszeichenkette Variables Zeichenkettensuffix Festes Zeichenkettensuffix

Hinweis: Sie können echte Sprachen zum Testen verwenden, aber in Weblate gibt es spezielle Pseudolokalisierungen – `en_XA` und `ar_XB`.

Hinweis: Sie können diese Erweiterung verwenden, um die Übersetzung in ein neues Gebietsschema einer bestehenden oder ähnlichen Sprache zu starten. Sobald Sie die Übersetzung zur Komponente hinzugefügt haben, folgen Sie der Erweiterung. *Beispiel:* Wenn Sie `fr` haben und die Übersetzung `fr_CA` starten wollen, setzen Sie einfach `fr` als Quelle, `fr_CA` als Ziel und lassen Sie das Präfix und Suffix leer.

Deinstallieren Sie die Erweiterung, sobald Sie die neue Übersetzung ausgefüllt haben, um zu verhindern, dass Weblate die nach dem Kopieren erstellten Übersetzungen verändert.

Mitwirkende im Kommentar

Erweiterungs-ID

`weblate.gettext.authors`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

Repository Pre-Commit

Aktualisiert den Kommentarteil in der Kopfzeile der PO-Datei, um den Namen des Mitwirkenden und das Datum des Beitrags aufzunehmen.

Die Kopfzeile der PO-Datei wird wie folgt aussehen:

```
# Michal Čihař <michal@weblate.org>, 2012, 2018, 2019, 2020.
# Pavel Borecki <pavel@example.com>, 2018, 2019.
# Filip Hron <filip@example.com>, 2018, 2019.
# anonymous <noreply@weblate.org>, 2019.
```

Aktualisieren Sie die Variable ALL_LINGUAS in der „configure“-Datei

Erweiterungs-ID

`weblate.gettext.configure`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

repository post-add, daily

Aktualisiert die ALL_LINGUAS-Variablen in `configure`, `configure.in` oder einer `configure.ac`-Datei, wenn eine neue Übersetzung hinzugefügt wird.

Ausgabe von Gettext anpassen

Erweiterungs-ID

`weblate.gettext.customize`

Konfiguration

<code>width</code>	Umbruch	Standardmäßig bricht gettext Zeilen mit 77 Zeichen und bei Zeilenvorschüben um. Mit dem Parameter <code>--no-wrap</code> wird der Umbruch nur bei Zeilenvorschub durchgeführt.
<code>language</code>		Verfügbare Auswahlmöglichkeiten:
<code>Zeilen</code>		77 – Zeilenumbruch nach 77 Zeichen und bei Zeilenumbrüchen (xgettext-Standard)
		65535 – Zeilenumbruch nur bei Zeilenumbrüchen (wie <code>xgettext --no-wrap</code>)
		-1 – Kein Zeilenumbruch

Trigger

storage post-load

Ermöglicht die Anpassung des Ausgabeverhaltens von Gettext, z. B. Zeilenumbrüche.

Es bietet die folgenden Optionen:

- Zeilenumbruch nach 77 Zeichen und bei Zeilenumbrüchen
- Zeilenumbruch nur nach Zeilenvorschüben

- Kein Umbruch langer Zeilen

Bemerkung: Standardmäßig bricht gettext Zeilen bei 77 Zeichen und bei Zeilenvorschüben um. Mit dem Parameter `--no-wrap` wird der Umbruch nur bei Zeilenvorschüben durchgeführt.

LINGUAS-Datei aktualisieren

Erweiterungs-ID

`weblate.gettext.linguas`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

repository post-add, daily

Aktualisiert die LINGUAS-Datei, wenn eine neue Übersetzung hinzugefügt wird.

MO-Dateien erzeugen

Erweiterungs-ID

`weblate.gettext.mo`

Konfiguration

<code>path</code>	Pfad der erzeugten MO-Datei	Wenn nicht angegeben, wird der Ort der PO-Datei verwendet.
-------------------	-----------------------------	--

Trigger

Repository Pre-Commit

Erzeugt automatisch eine MO-Datei für jede geänderte PO-Datei.

Der Speicherort der erzeugten MO-Datei kann angepasst werden, und das Feld dafür verwendet *Template markup*.

PO-Dateien auf POT aktualisieren (msgmerge)

Erweiterungs-ID

`weblate.gettext.msgmerge`

Konfiguration

<code>previous</code>	Frühere msgids der übersetzten Zeichenketten beibehalten
<code>no_location</code>	Speicherorte der übersetzten Zeichenketten entfernen
<code>fuzzy</code>	Fuzzy-Matching verwenden

Trigger

Repository nach dem Update

Aktualisiert alle PO-Dateien (wie durch *Dateimasken* konfiguriert), damit sie mit der POT-Datei (wie durch *Vorlage für neue Übersetzungen* konfiguriert) übereinstimmen, unter Verwendung von **msgmerge**.

Wird immer dann ausgelöst, wenn neue Änderungen aus dem Upstream-Repository übernommen werden. Die meisten msgmerge-Befehlszeilenooptionen können über die Konfiguration der Erweiterung eingerichtet werden.

Siehe auch:

Aktualisiert Weblate neben den Übersetzungen auch die Übersetzungsdateien?

Git-Commits zusammenfassen

Erweiterungs-ID

`weblate.git.squash`

Konfiguration

<code>squash</code>	Commits zusammenfassen	Verfügbare Auswahlmöglichkeiten: <code>all</code> – Alle Commits in einem <code>language</code> – Pro Sprache <code>file</code> – Pro Datei <code>author</code> – Pro Autor
<code>append_tr</code>	Endzeilen der zusammengefassten Commit-Nachricht hinzufügen	Die Endzeilen ähneln RFC-822-E-Mail-Headern, am Ende eines ansonsten freien Teils der Commit-Nachricht, z. B. „Mitverfasst von: ...“.
<code>commit_mes</code>	Commit-Nachricht	Diese Commit-Nachricht wird anstelle der zusammengefassten Commit-Nachrichten verwendet.

Trigger

`repository post-commit`

Git-Commits vor dem Pushen der Änderungen zusammenfassen.

Git-Commits können vor dem Pushen von Änderungen in einem der folgenden Modi zusammengefasst werden:

- Alle Commits in einen
- Pro Sprache
- Pro Datei
- Pro Übersetzer

Ursprüngliche Commit-Nachrichten werden erhalten, verlieren aber die Urheberschaft, es sei denn *Pro Autor* wird ausgewählt oder die Commit-Nachricht wird angepasst, um sie einzubeziehen.

Die ursprünglichen Commit-Nachrichten können optional mit einer eigenen Commit-Nachricht überschrieben werden.

Endzeilen (Commit-Zeilen wie `Co-authored-by: ...`) können optional aus den ursprünglichen Commit-Nachrichten entfernt und an das Ende der zusammengefassten Commit-Nachricht angehängt werden. Dies erzeugt auch eine angemessene `Co-authored-by: -Anerkennung` für jeden Übersetzer.

Ausgabe von JSON anpassen

Erweiterungs-ID

`weblate.json.customize`

Konfiguration

<code>sort_key</code>	Nach JSON-Schlüssel sortieren	
<code>indent</code>	JSON-Einrückung	
<code>style</code>	JSON-Einrückungsstil	Verfügbare Auswahlmöglichkeiten: <code>spaces</code> – Leerzeichen <code>tabs</code> – Tabulatoren

Trigger

`storage post-load`

Ermöglicht die Anpassung des JSON-Ausgabeverhaltens, z. B. Einrückung und Sortierung.

Die Übersetzung der Java-Eigenschaften-Datei formatieren

Erweiterungs-ID

`weblate.properties.sort`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

Repository Pre-Commit

Formatiert und sortiert die Java-Eigenschaftendatei.

- Konsolidiert Zeilenumbrüche zu Unix-Zeilenumbrüchen.
- Großbuchstabenformatierung von Unicode-Escape-Sequenzen (falls vorhanden).
- Entfernt leere Zeilen und Kommentare.
- Sortiert die Zeichenketten nach Schlüsselworten.
- Bereinigt doppelte Zeichenketten.

Veralteten Kommentar entfernen

Neu in Version 3.7.

Erweiterungs-ID

`weblate.removal.comments`

Konfiguration

age	Tage aufzubewahren
-----	--------------------

Trigger

täglich

Legen Sie einen Zeitrahmen für die Entfernung von Kommentaren fest.

Dies kann nützlich sein, um alte Kommentare zu entfernen, die möglicherweise nicht mehr aktuell sind. Seien Sie vorsichtig, denn wenn Kommentare alt werden, bedeutet das nicht, dass sie ihre Bedeutung verloren haben.

Entfernen von veraltetem Vorschlag

Neu in Version 3.7.

Erweiterungs-ID

`weblate.removal.suggestions`

Konfiguration

age	Tage aufzubewahren	
votes	Abstimmungsschwelle	Schwellwert für die Entfernung. Dieses Feld hat keine Auswirkung, wenn die Abstimmung deaktiviert ist.

Trigger

täglich

Legen Sie einen Zeitrahmen für die Entfernung von Vorschlägen fest.

Kann im Zusammenhang mit Übersetzungsvorschlägen (siehe [Peer-Review](#)) sehr nützlich sein, um Vorschläge zu entfernen, die innerhalb eines bestimmten Zeitraums nicht genügend positive Stimmen erhalten haben.

RESX-Dateien aktualisieren

Neu in Version 3.9.

Erweiterungs-ID

`weblate.resx.update`

Konfiguration

Diese Erweiterung hat keine Konfiguration.

Trigger

Repository nach dem Update

Aktualisieren Sie alle Übersetzungsdateien entsprechend der einsprachigen Upstream-Basisdatei. Nicht verwendete Zeichenketten werden entfernt und neue Zeichenketten werden als Kopien der Ausgangszeichenkette hinzugefügt.

Hinweis: Verwenden Sie [Übersetzungsdateien bereinigen](#), wenn Sie nur veraltete Übersetzungsschlüssel entfernen wollen.

Siehe auch:

Aktualisiert Weblate neben den Übersetzungen auch die Übersetzungsdateien?

XML-Ausgabe anpassen

Neu in Version 4.15.

Erweiterungs-ID

`weblate.xml.customize`

Konfiguration

<code>closing_tags</code>	Schließendes Tag für leere XML-Tags einschließen
---------------------------	--

Trigger

storage post-load

Ermöglicht die Anpassung des XML-Ausgabeverhaltens, z. B. das Schließen von Tags anstelle von selbstschließenden Tags für leere Tags.

YAML-Ausgabe anpassen

Neu in Version 3.10.2.

Erweiterungs-ID

`weblate.yaml.customize`

Konfiguration

indent	YAML Einrückung	
width	Umlang bruch Zeil Zeil	Verfügbare Auswahlmöglichkeiten: 80 – Zeilenumbruch bei 80 Zeichen 100 – Zeilenumbruch bei 100 Zeichen 120 – Zeilenumbruch bei 120 Zeichen 180 – Zeilenumbruch bei 180 Zeichen 65535 – Kein Zeilenumbruch
line_break	Zeilenumbrüche	Verfügbare Auswahlmöglichkeiten: dos – DOS (\r\n) unix – UNIX (\n) mac – MAC (\r)

Trigger

storage post-load

Ermöglicht die Anpassung des YAML-Ausgabeverhaltens, z. B. Zeilenlänge oder Zeilenvorschübe.

2.14.2 Liste der Erweiterungen anpassen

Die Liste der Erweiterungen wird durch `WEBLATE_ADDONS` konfiguriert. Um eine weitere Erweiterung hinzuzufügen, geben Sie einfach den absoluten Klassennamen in dieser Einstellung an.

2.14.3 Schreiben einer Erweiterung

Sie können auch Ihre eigenen Erweiterungen schreiben, eine Unterklasse von `weblate.addons.base.BaseAddon` erstellen, um die Metadaten der Erweiterung zu definieren, und dann einen Callback implementieren, um die Verarbeitung durchzuführen.

Siehe auch:

Erweiterungen entwickeln

2.14.4 Ausführen von Skripten der Erweiterung

Erweiterungen können auch verwendet werden, um externe Skripte auszuführen. Früher war dies in Weblate integriert, aber jetzt müssen Sie etwas Code schreiben, um Ihr Skript mit einer Erweiterung zu umschließen.

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Example pre commit script."""

from django.utils.translation import gettext_lazy as _

from weblate.addons.events import EVENT_PRE_COMMIT
from weblate.addons.scripts import BaseScriptAddon

class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
name = "weblate.example.pre"
# Verbose name and long description
verbose = _("Execute script before commit")
description = _("This add-on executes a script.")

# Script to execute
script = "/bin/true"
# File to add in commit (for pre commit event)
# does not have to be set
add_file = "po/{{ language_code }}.po"
```

Für Installationsanweisungen siehe *Benutzerdefinierte Qualitätsprüfungen, Erweiterungen und automatische Korrekturen*.

Das Skript wird ausgeführt, wobei das aktuelle Verzeichnis auf das Stammverzeichnis des VCS-Repositorys für eine beliebige Komponente gesetzt wird.

Zusätzlich sind die folgenden Umgebungsvariablen verfügbar:

WL_VCS

Versionsverwaltung verwendet.

WL_REPO

URL des Upstream-Repositorys.

WL_PATH

Absoluter Pfad zum VCS-Repository.

WL_BRANCH

Neu in Version 2.11.

In der aktuellen Komponente konfigurierter Repository-Branch.

WL_FILEMASK

Dateimaske für die aktuelle Komponente.

WL_TEMPLATE

Dateiname der Vorlage für einsprachige Übersetzungen (kann leer sein).

WL_NEW_BASE

Neu in Version 2.14.

Dateiname der Datei, die für die Erstellung neuer Übersetzungen verwendet wird (kann leer sein).

WL_FILE_FORMAT

In der aktuellen Komponente verwendetes Dateiformat.

WL_LANGUAGE

Sprache der aktuell bearbeiteten Übersetzung (nicht verfügbar für Hooks auf Komponentenebene).

WL_PREVIOUS_HEAD

Vorheriger HEAD nach der Aktualisierung (nur verfügbar nach Ausführung des Post-Update-Hooks).

WL_COMPONENT_SLUG

Neu in Version 3.9.

Komponenten-Slug, der zum Aufbau der URL verwendet wird.

WL_PROJECT_SLUG

Neu in Version 3.9.

Projekt-Slug, der zum Aufbau der URL verwendet wird.

WL_COMPONENT_NAME

Neu in Version 3.9.

Name der Komponente.

WL_PROJECT_NAME

Neu in Version 3.9.

Projektname.

WL_COMPONENT_URL

Neu in Version 3.9.

URL der Komponente.

WL_ENGAGE_URL

Neu in Version 3.9.

URL des beteiligten Projekts.

Siehe auch:

Komponentenkonfiguration

Repository-Verarbeitung nach Aktualisierung

Kann verwendet werden, um Übersetzungsdateien zu aktualisieren, wenn sich der Upstream-Quellcode des VCS ändert. Um dies zu erreichen, denken Sie bitte daran, dass Weblate nur Dateien sieht, die an das VCS übertragen wurden, sodass Sie die Änderungen als Teil des Skripts committen müssen.

Zum Beispiel mit Gulp können Sie es mit folgendem Code machen:

```
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Verarbeitung von Übersetzungen vor dem Commit

Verwenden Sie das Commit-Skript, um eine Übersetzung automatisch zu ändern, bevor sie in das Repository übertragen wird.

Sie wird als einzelner Parameter übergeben, der aus dem Dateinamen der aktuellen Übersetzung besteht.

2.15 Übersetzungsspeicher

Neu in Version 2.20.

Weblate comes with a built-in translation memory consisting of the following:

- Manually imported translation memory (see *Benutzeroberfläche*).
- Automatically stored translations performed in Weblate (depending on *Translation memory scopes*).
- Automatically imported past translations.

Content in the translation memory can be applied one of two ways:

- Manually, *Automatische Vorschläge* view while translating.
- Automatically, by translating strings using *Automatische Übersetzung*, or *Automatische Übersetzung* add-on.

For installation tips, see *Weblate Translation Memory*, which is turned on by default.

2.15.1 Translation memory scopes

Neu in Version 3.2: In earlier versions translation memory could be only loaded from a file corresponding to the current imported translation memory scope.

The translation memory scopes are there to allow both privacy and sharing of translations, to suit the desired behavior.

Importierter Übersetzungsspeicher

Importing arbitrary translation memory data using the `import_memory` command makes memory content available to all users and projects.

Übersetzungsspeicher pro Benutzer

Stores all user translations automatically in the personal translation memory of each respective user.

Übersetzungsspeicher pro Projekt

All translations within a project are automatically stored in a project translation memory only available for this project.

Geteilter Übersetzungsspeicher

All translation within projects with shared translation memory turned on are stored in a shared translation memory available to all projects.

Please consider carefully whether to turn this feature on for shared Weblate installations, as it can have severe implications:

- The translations can be used by anybody else.
- This might lead to disclosing secret information.

2.15.2 Übersetzungsspeicher verwalten

Benutzeroberfläche

Neu in Version 3.2.

In the basic user interface you can manage per user and per project translation memories. It can be used to download, wipe or import translation memory.

Hinweis: Translation memory in JSON can be imported into Weblate, TMX is provided for interoperability with other tools.

Siehe auch:

Weblate Übersetzungsspeicher-Schema

Verwaltungsoberfläche

There are several management commands to manipulate the translation memory content. These operate on the translation memory as whole, unfiltered by scopes (unless requested by parameters):

`dump_memory`

Exports the memory into JSON

`import_memory`

Imports TMX or JSON files into the translation memory

2.16 Konfiguration

Alle Einstellungen werden in `settings.py` gespeichert (wie bei Django üblich).

Bemerkung: Nachdem Sie eine dieser Einstellungen geändert haben, müssen Sie Weblate neu starten – sowohl den WSGI- als auch den Celery-Prozess.

Wenn es als `mod_wsgi` ausgeführt wird, müssen Sie Apache neu starten, um die Konfiguration neu zu laden.

Siehe auch:

Bitte lesen Sie auch die [Django-Dokumentation](#) für Parameter zur Konfiguration von Django selbst.

2.16.1 AKISMET_API_KEY

Weblate kann Akismet verwenden, um eingehende anonyme Vorschläge auf Spam zu überprüfen. Besuchen Sie akismet.com, um einen API-Schlüssel zu erwerben und ihn mit einer Website zu verknüpfen.

2.16.2 ANONYMOUS_USER_NAME

Benutzername von Benutzern, die nicht angemeldet sind.

Siehe auch:

Zugriffssteuerung

2.16.3 AUDITLOG_EXPIRY

Neu in Version 3.6.

Wie viele Tage Weblate Audit-Protokolle aufbewahren soll, die Informationen über Kontoaktivitäten enthalten.

Der Standardwert ist 180 Tage.

2.16.4 AUTH_LOCK_ATTEMPTS

Neu in Version 2.14.

Maximale Anzahl der fehlgeschlagenen Authentifizierungsversuche, bevor die Ratenbegrenzung angewendet wird.

Dies wird derzeit an den folgenden Standorten angewandt:

- Anmelden. Löscht das Kennwort des Kontos und verhindert, dass sich der Benutzer anmelden kann, ohne ein neues Kennwort anzufordern.
- Passwort zurücksetzen. Verhindert, dass neue E-Mails gesendet werden, damit die Benutzer nicht mit zu vielen Versuchen, ihr Passwort zurückzusetzen, belästigt werden.

Der Standardwert ist 10.

Siehe auch:

Ratenbegrenzung

2.16.5 AUTO_UPDATE

Neu in Version 3.2.

Geändert in Version 3.11: Die ursprüngliche Option „Ein/Aus“ wurde geändert, um zu unterscheiden, welche Zeichenketten akzeptiert werden.

Aktualisiert täglich alle Repositories.

Hinweis: Nützlich, wenn Sie *Benachrichtigungs-Hooks* nicht verwenden, um Weblate-Repositories automatisch zu aktualisieren.

Bemerkung: Aus Gründen der Abwärtskompatibilität gibt es neben der Auswahl der Zeichenkette auch Ein/Aus-Optionen.

Die Optionen sind:

„none“

Keine täglichen Aktualisierungen.

„remote“ auch `` False

Aktualisieren Sie nur Remotes.

„full“ auch `` True

Aktualisieren Sie Remotes und führen Sie Arbeitskopien zusammen.

Bemerkung: Dies setzt voraus, dass *Hintergrundaufgaben mit Celery* funktioniert, und wird nach einem Neustart wirksam.

2.16.6 AVATAR_URL_PREFIX

Präfix für den Aufbau von Avatar-URLs als: „`${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`“. Die folgenden Dienste sind dafür bekannt, dass sie funktionieren:

Gravatar (Standard), wie unter <https://gravatar.com/>
`AVATAR_URL_PREFIX = 'https://www.gravatar.com/'`

Libravatar, wie unter <https://www.libravatar.org/>
`AVATAR_URL_PREFIX = 'https://www.libravatar.org/'`

Siehe auch:

Avatar-Caching, *ENABLE_AVATARS*, *Avatars*

2.16.7 AUTH_TOKEN_VALID

Neu in Version 2.14.

Dauer der Gültigkeit des Authentifizierungstokens und des temporären Kennworts aus E-Mails zum Zurücksetzen des Kennworts. Wird in Sekunden angegeben, Standardwert ist 172800 (2 Tage).

2.16.8 AUTH_PASSWORD_DAYS

Neu in Version 2.15.

Für wie viele Tage Weblate die Wiederverwendung eines zuvor verwendeten Passworts für einen Benutzer ablehnt.

Die Prüfung basiert auf dem Audit-Protokoll, *AUDITLOG_EXPIRY* muss mindestens so hoch sein wie diese Einstellung.

Bemerkung: Passwortänderungen, die vor Weblate 2.15 vorgenommen wurden, werden in dieser Richtlinie nicht berücksichtigt.

Der Standardwert ist 180 Tage.

2.16.9 AUTOFIX_LIST

Liste der automatischen Korrekturen, die beim Speichern einer Zeichenkette anzuwenden sind.

Bemerkung: Geben Sie einen vollqualifizierten Pfad zu der Python-Klasse an, die die Autofixer-Schnittstelle implementiert.

Verfügbare Korrekturen:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace`
Passt Leerzeichen am Anfang und Ende der Zeichenkette an die Ausgangszeichenkette an.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis`
Ersetzt nachgestellte Punkte (...), wenn die Ausgangszeichenkette entsprechende Auslassungspunkte (...) enthält.

weblate.trans.autofixes.chars.RemoveZeroSpace

Entfernt Leerzeichen ohne Breite, wenn die Ausgangszeichenkette keine enthält.

weblate.trans.autofixes.chars.RemoveControlChars

Entfernt Steuerzeichen, wenn die Ausgangszeichenkette keine enthält.

weblate.trans.autofixes.html.BleachHTML

Entfernt unsichere HTML-Auszeichnungen aus Zeichenketten, die als `safe-html` gekennzeichnet sind (siehe *Unsicheres HTML*).

Sie können auswählen, welche verwendet werden sollen:

```
AUTOFIX_LIST = (  
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",  
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",  
)
```

Siehe auch:

Automatische Korrekturen, Benutzerdefinierte automatische Korrekturen

2.16.10 BACKGROUND_TASKS

Neu in Version 4.5.2.

Legt fest, wie oft langwierige Wartungsmaßnahmen für eine Komponente ausgelöst werden sollen.

Im Moment steuert dies:

- *Automatische Übersetzung* Erweiterung
- *Überprüfungen und Korrekturen* Neuberechnung

Mögliche Auswahlmöglichkeiten:

- *monthly*“ (dies ist die Standardeinstellung)
- *weekly*
- *daily*
- *never*

Bemerkung: Es wird nicht empfohlen, die Häufigkeit zu erhöhen, wenn Weblate Tausende von Komponenten enthält.

2.16.11 BASIC_LANGUAGES

Neu in Version 4.4.

Liste der Sprachen, die dem Benutzer zum Starten einer neuen Übersetzung angeboten wird. Wenn nicht angegeben, wird die integrierte Liste mit allen häufig verwendeten Sprachen verwendet, jedoch ohne länderspezifische Varianten.

Dies schränkt nur nicht privilegierte Benutzer ein, unerwünschte Sprachen hinzuzufügen. Den Projektadministratoren steht nach wie vor die volle Auswahl der in Weblate definierten Sprachen zur Verfügung.

Bemerkung: Damit werden keine neuen Sprachen für Weblate definiert, sondern nur die in der Datenbank vorhandenen gefiltert.

Beispiel:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Siehe auch:

Sprachdefinitionen

2.16.12 BORG_EXTRA_ARGS

Neu in Version 4.9.

Sie können zusätzliche Argumente an **borg create** übergeben, wenn integrierte Backups ausgelöst werden.

Beispiel:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Siehe auch:

Sichern und Verschieben von Weblate, borg create

2.16.13 CACHE_DIR

Neu in Version 4.16.

Das Verzeichnis, in dem Weblate Cache-Dateien speichert. Standardmäßig ist dies der Unterordner `cache` in `DATA_DIR`.

Ändern Sie dies in lokales oder temporäres Dateisystem, wenn `DATA_DIR` auf einem Netzwerk-Dateisystem liegt.

Der Docker-Container verwendet dafür ein eigenes Volume, siehe *Docker-Container-Volumes*.

2.16.14 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Passen Sie den Header `Content-Security-Policy` für Weblate an. Die Kopfzeile wird automatisch auf der Grundlage der aktivierten Integrationen mit Diensten von Drittanbietern (Matomo, Google Analytics, Sentry, ...) generiert.

Alle diese Listen sind standardmäßig leer.

Beispiel:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Siehe auch:

Richtlinie zur Inhaltssicherheit, Content Security Policy (CSP)

2.16.15 CHECK_LIST

Liste der an einer Übersetzung durchzuführenden Qualitätsprüfungen.

Bemerkung: Geben Sie einen vollqualifizierten Pfad zu der Python-Klasse an, die die Prüfschnittstelle implementiert.

Passen Sie die Liste der Qualitätsprüfungen an die für Sie relevanten Prüfungen an.

Alle integrierten Qualitätsprüfungen sind standardmäßig aktiviert, von wo aus Sie diese Einstellungen ändern können. Standardmäßig sind sie in Muster-Konfiguration auskommentiert, sodass Standardwerte verwendet werden. Für jede neue Weblate-Version werden dann neue Qualitätsprüfungen durchgeführt.

Sie können alle Qualitätsprüfungen deaktivieren:

```
CHECK_LIST = ()
```

Sie können nur ein paar einschalten:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Bemerkung: Eine Änderung dieser Einstellung wirkt sich nur auf neu geänderte Übersetzungen aus, bestehende Qualitätsprüfungen werden weiterhin in der Datenbank gespeichert. Um auch Änderungen an den gespeicherten Übersetzungen vorzunehmen, führen Sie *updatechecks* aus.

Siehe auch:

Qualitätsprüfungen, Anpassen des Verhaltens mit Markierungen

2.16.16 COMMENT_CLEANUP_DAYS

Neu in Version 3.6.

Löscht Kommentare nach einer bestimmten Anzahl von Tagen. Der Standardwert ist `None`, was bedeutet, dass überhaupt nicht gelöscht wird.

2.16.17 COMMIT_PENDING_HOURS

Neu in Version 2.10.

Anzahl der Stunden zwischen Commits von ausstehenden Änderungen durch die Hintergrundaufgabe.

Siehe auch:

Komponentenkonfiguration, Alter der Änderungen, das erreicht sein muss, bevor ein Commit erfolgt, Ausführen von Wartungsaufgaben, `commit_pending`

2.16.18 CONTACT_FORM

Neu in Version 4.6.

Legt fest, wie E-Mails aus dem Kontaktformular gesendet werden. Wählen Sie eine Konfiguration, die Ihrer Mail-Server-Konfiguration entspricht.

"reply-to"

Der Absender wird in *Reply-To* verwendet, dies ist das Standardverhalten.

"from"

Der Absender wird in *From* verwendet. Ihr Mail-Server muss den Versand solcher E-Mails erlauben.

2.16.19 DATA_DIR

Der Ordner, in dem Weblate alle Daten speichert. Er enthält Links zu VCS-Repositorys, einen Volltextindex und verschiedene Konfigurationsdateien für externe Tools.

Üblicherweise existieren folgende Unterverzeichnisse:

home

Basisverzeichnis, das zum Aufrufen von Skripts verwendet wird.

ssh

SSH-Schlüssel und Konfiguration.

static

Standardspeicherort für statische Django-Dateien, angegeben durch `STATIC_ROOT`. Siehe *Bereitstellung statischer Dateien*.

Der Docker-Container verwendet dafür ein eigenes Volume, siehe *Docker-Container-Volumes*.

media

Standardspeicherort für Django-Mediendateien, angegeben durch `MEDIA_ROOT`. Enthält hochgeladene Bildschirmfotos, siehe *Bildschirmfotos*.

vcs

Versionsverwaltung-Repositorys für Übersetzungen.

backups

Tägliche Sicherungsdaten, siehe *Gedumpte Daten für Backups* für Details.

fonts:

Vom Benutzer hochgeladene Schriftarten, siehe *Schriftarten verwalten*.

cache

Verschiedene Caches können mit `CACHE_DIR` an anderer Stelle platziert werden.

Der Docker-Container verwendet dafür ein eigenes Volume, siehe *Docker-Container-Volumes*.

Bemerkung: Dieses Verzeichnis muss von Weblate beschreibbar sein. Die Ausführung als uWSGI bedeutet, dass der Benutzer `www-data` Schreibzugriff darauf haben sollte.

Der einfachste Weg, dies zu erreichen, ist, den Benutzer zum Eigentümer des Verzeichnisses zu machen:

```
sudo chown www-data:www-data -R $DATA_DIR
```

Die Standardeinstellung ist `/home/weblate/data`, aber es wird erwartet, dass sie konfiguriert wird.

Siehe auch:

Dateisystemberechtigungen, Sichern und Verschieben von Weblate, `CACHE_DIR`

2.16.20 DATABASE_BACKUP

Neu in Version 3.1.

Ob die Datenbank-Backups als reiner Text oder komprimiert gespeichert oder übersprungen werden sollen, die zulässigen Werte sind:

- "plain"
- "komprimiert"
- „none“

Siehe auch:

Sichern und Verschieben von Weblate

2.16.21 DEFAULT_ACCESS_CONTROL

Neu in Version 3.3.

Die Standardeinstellung der Zugriffssteuerung für neue Projekte:

- 0
Öffentlich
- 1
Geschützt
- 100
Privat
- 200
Benutzerdefiniert

Verwenden Sie *Benutzerdefiniert*, wenn Sie die ACL manuell verwalten, d. h. sich nicht auf die interne Weblate-Verwaltung verlassen.

Siehe auch:

Projekt-Zugriffssteuerung, Zugriffssteuerung

2.16.22 DEFAULT_AUTO_WATCH

Neu in Version 4.5.

Konfiguriert, ob *Projekte nach Beitrag automatisch beobachten* für neue Benutzer aktiviert werden soll. Der Standardwert ist `True`.

Siehe auch:

Benachrichtigungen

2.16.23 DEFAULT_RESTRICTED_COMPONENT

Neu in Version 4.1.

Der Standardwert für die Komponenteneinschränkung.

Siehe auch:

Eingeschränkter Zugriff, Umfang der Teams

2.16.24 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DE- FAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DE- FAULT_MERGE_MESSAGE

Voreingestellte Commit-Nachrichten für verschiedene Operationen, siehe *Komponentenkonfiguration* für Details.

Siehe auch:

Template markup, Komponentenkonfiguration, Commit-, Add-, Delete-, Merge-, Erweiterungs- und Merge-Request-Nachrichten

2.16.25 DEFAULT_ADDONS

Standard-Erweiterungen, die für jede erstellte Komponente installiert werden.

Bemerkung: Diese Einstellung wirkt sich nur auf neu erstellte Komponenten aus.

Beispiel:

```
DEFAULT_ADDONS = {
    # Add-on with no parameters
    "weblate.flags.target_edit": {},
    # Add-on with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

Siehe auch:

install_addon, Erweiterungen, WEBLATE_ADDONS

2.16.26 DEFAULT_COMMITER_EMAIL

Neu in Version 2.4.

Die E-Mail-Adresse des Committers ist standardmäßig auf `noreply@weblate.org` eingestellt.

Siehe auch:

DEFAULT_COMMITER_NAME

2.16.27 DEFAULT_COMMITER_NAME

Neu in Version 2.4.

Der Name des Committers ist standardmäßig `Weblate`.

Siehe auch:

DEFAULT_COMMITER_EMAIL

2.16.28 DEFAULT_LANGUAGE

Neu in Version 4.3.2.

Standard-Ausgangssprache, die zum Beispiel in *Ausgangssprache* zu verwenden ist.

Der Standardwert ist „en“. Das entsprechende Sprachobjekt muss in der Datenbank vorhanden sein.

Siehe auch:

Sprachdefinitionen, Ausgangssprache

2.16.29 DEFAULT_MERGE_STYLE

Neu in Version 3.4.

Merge-Stil für alle neuen Komponenten.

- *rebase* – Standard
- *merge*

Siehe auch:

Komponentenkonfiguration, Git-Strategie

2.16.30 DEFAULT_SHARED_TM

Neu in Version 3.2.

Konfiguriert den Standardwert von *Gemeinsamen Übersetzungsspeicher verwenden* und *Zu einem gemeinsamen Übersetzungsspeicher beitragen*.

2.16.31 DEFAULT_TRANSLATION_PROPAGATION

Neu in Version 2.5.

Standardeinstellung für die Übersetzungsweitergabe, ist standardmäßig „True“.

Siehe auch:

Komponentenkonfiguration, Verbreitung von Übersetzungen erlauben

2.16.32 DEFAULT_PULL_MESSAGE

Konfiguriert den Standardtitel und die Standardnachricht für Pull Requests.

2.16.33 ENABLE_AVATARS

Ob Gravatar-basierte Avatare für Benutzer aktiviert werden sollen. Standardmäßig ist dies aktiviert.

Avatare werden abgerufen und auf dem Server zwischengespeichert, wodurch das Risiko des Durchsickerns privater Informationen verringert und das Benutzererlebnis beschleunigt wird.

Siehe auch:

Avatar-Caching, AVATAR_URL_PREFIX, Avatars

2.16.34 ENABLE_HOOKS

Ob anonyme Remote-Hooks aktiviert werden sollen.

Siehe auch:

Benachrichtigungs-Hooks

2.16.35 ENABLE_HTTPS

Ob Links per HTTPS oder HTTP an Weblate gesendet werden sollen. Diese Einstellung wirkt sich auf gesendete E-Mails und generierte absolute URLs aus.

In der Grundeinstellung wird dies auch für viele Django-Einstellungen im Rahmen von HTTPS verwendet – es aktiviert sichere Cookies, schaltet HSTS um oder aktiviert die Weiterleitung zu einer HTTPS-URL.

Die HTTPS-Umleitung könnte in einigen Fällen problematisch sein und Sie könnten auf das Problem einer unendlichen Umleitung stoßen, wenn Sie einen Reverse-Proxy verwenden, der die SSL-Terminierung vornimmt und die Protokoll-Header nicht korrekt an Django weitergibt. Bitte passen Sie Ihre Reverse-Proxy-Konfiguration so an, dass X-Forwarded-Proto- oder Forwarded-Header ausgegeben werden oder konfigurieren Sie `SECURE_PROXY_SSL_HEADER`, damit Django den SSL-Status korrekt erkennt.

Siehe auch:

`SESSION_COOKIE_SECURE`, `CSRF_COOKIE_SECURE`, `SECURE_SSL_REDIRECT`, `SECURE_PROXY_SSL_HEADER` *Seitendomain richtig einstellen*

2.16.36 ENABLE_SHARING

Schalten Sie das Menü *Teilen* ein oder aus, damit Benutzer den Übersetzungsfortschritt in sozialen Netzwerken teilen können.

2.16.37 EXTRA_HTML_HEAD

Neu in Version 4.15.

Fügen Sie zusätzliches Markup in den HTML-Header ein. Kann zum Beispiel für die Überprüfung der Eigentümerschaft einer Website verwendet werden:

```
EXTRA_HTML_HEAD = '<link href="https://fosstodon.org/@weblate" rel="me">'
```

Warnung: Die Zeichenfolge wird nicht bereinigt, sondern unverändert in den HTML-Header eingefügt.

2.16.38 GET_HELP_URL

Neu in Version 4.5.2.

URL, unter der Unterstützung für Ihre Weblate-Instanz zu finden ist.

2.16.39 GITEA_CREDENTIALS

Neu in Version 4.12.

Liste der Zugangsdaten für Gitea-Server.

```
GITEA_CREDENTIALS = {
  "try.gitea.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitea.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Siehe auch:

Gitea-Pull-Requests, Erstellen eines persönlichen Gitea-Zugangstokens

2.16.40 GITLAB_CREDENTIALS

Neu in Version 4.3.

Liste der Zugangsdaten für GitLab-Server.

```
GITLAB_CREDENTIALS = {
  "gitlab.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitlab.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Siehe auch:

GitLab-Merge-Requests, GitLab: Persönliche Zugangs-Token

2.16.41 GITHUB_CREDENTIALS

Neu in Version 4.3.

Liste für Zugangsdaten für GitHub-Server.

```
GITHUB_CREDENTIALS = {
  "api.github.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "github.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Siehe auch:

GitHub-Pull-Requests, Erstellen eines persönlichen GitHub-Zugangstokens

2.16.42 BITBUCKETSERVER_CREDENTIALS

Neu in Version 4.16.

Liste der Zugangsdaten für Bitbucket-Server.

```
BITBUCKETSERVER_CREDENTIALS = {  
    "git.self-hosted.com": {  
        "username": "weblate",  
        "token": "http-access-token",  
    },  
}
```

Siehe auch:

Bitbucket Server pull requests, *Bitbucket: HTTP-Zugangstoken*

2.16.43 GOOGLE_ANALYTICS_ID

Google Analytics-ID, um die Überwachung von Weblate mit Google Analytics zu aktivieren.

2.16.44 HIDE_REPO_CREDENTIALS

Ausblenden der Repository-Zugangsdaten auf der Weboberfläche. Falls Sie eine Repository-URL mit Benutzer und Passwort haben, wird Weblate diese ausblenden, wenn zugehörige Informationen den Benutzern angezeigt werden.

Zum Beispiel wird anstelle von `https://Benutzer:Passwort@git.beispiel.de/repo.git` nur `https://git.beispiel.de/repo.git` angezeigt. In ähnlicher Weise versucht es auch, VCS-Fehlermeldungen zu bereinigen.

Bemerkung: Diese Funktion ist standardmäßig eingeschaltet.

2.16.45 HIDE_VERSION

Neu in Version 4.3.1.

Versteckt Versionsinformationen vor nicht authentifizierten Benutzern. Dies führt auch dazu, dass alle Links zur Dokumentation auf die neueste Version verweisen, anstatt auf die Dokumentation, die der aktuell installierten Version entspricht.

Das Verstecken der Version ist in einigen Unternehmen eine empfohlene Sicherheitspraxis, aber es verhindert nicht, dass Angreifer die Version durch Ausprobieren des Verhaltens herausfinden können.

Bemerkung: Diese Funktion ist standardmäßig ausgeschaltet.

2.16.46 INTERLEDGER_PAYMENT_POINTERS

Neu in Version 4.12.1.

Liste der Interledger Payment Pointers (ILPs) für Web-Monetarisierung.

Wenn mehrere angegeben werden, wird eine probabilistische Aufteilung der Einnahmen durch zufällige Auswahl erreicht.

Weitere Einzelheiten finden Sie unter [<https://webmonetization.org/>](https://webmonetization.org/).

Hinweis: Mit dem Standardwert können Benutzer Weblate selbst finanzieren.

2.16.47 IP_BEHIND_REVERSE_PROXY

Neu in Version 2.14.

Gibt an, ob Weblate hinter einem Reverse-Proxy ausgeführt wird.

Wenn auf `True` gesetzt, bezieht Weblate die IP-Adresse aus einem Header, der durch `IP_PROXY_HEADER` definiert ist.

Warnung: Vergewissern Sie sich, dass Sie tatsächlich einen Reverse-Proxy verwenden und dass dieser diesen Header setzt, da die Benutzer sonst die IP-Adresse fälschen können.

Bemerkung: Diese Funktion ist nicht standardmäßig eingeschaltet.

Siehe auch:

Hinter einem Reverse-Proxy ausführen, Ratenbegrenzung, IP_PROXY_HEADER, IP_PROXY_OFFSET

2.16.48 IP_PROXY_HEADER

Neu in Version 2.14.

Gibt an, aus welchem Header Weblate die IP-Adresse beziehen soll, wenn `IP_BEHIND_REVERSE_PROXY` eingeschaltet ist.

Die Standardeinstellung ist `HTTP_X_FORWARDED_FOR`.

Siehe auch:

Hinter einem Reverse-Proxy ausführen, Ratenbegrenzung, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_OFFSET

2.16.49 IP_PROXY_OFFSET

Neu in Version 2.14.

Gibt an, welcher Teil von `IP_PROXY_HEADER` als Client-IP-Adresse verwendet wird.

Abhängig von Ihrer Einstellung kann dieser Header aus mehreren IP-Adressen bestehen (z. B. `X-Forwarded-For: a, b, client-ip`) und Sie können hier konfigurieren, welche Adresse aus dem Header als Client-IP-Adresse verwendet wird.

Warnung: Diese Einstellung wirkt sich auf die Sicherheit Ihrer Installation aus. Sie sollten sie nur so konfigurieren, dass vertrauenswürdige Proxys zur Ermittlung der IP-Adresse verwendet werden.

Der Standardwert ist 0.

Siehe auch:

Hinter einem Reverse-Proxy ausführen, Ratenbegrenzung, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER

2.16.50 LEGAL_TOS_DATE

Neu in Version 4.15.

Bemerkung: Sie müssen *Rechtliche Grundlagen* installiert haben, damit dies funktioniert.

Datum der letzten Aktualisierung der Dokumente zu den Nutzungsbedingungen. Immer, wenn sich das Datum ändert, müssen die Nutzer den Nutzungsbedingungen zustimmen.

```
from datetime import date

LEGAL_TOS_DATE = date(2022, 2, 2)
```

2.16.51 LEGAL_URL

Neu in Version 3.5.

URL, unter der Ihre Weblate-Instanz ihre juristischen Dokumente anzeigt.

Hinweis: Nützlich, wenn Sie Ihre juristischen Dokumente außerhalb von Weblate hosten, um sie in Weblate einzubetten. Bitte schauen Sie sich *Rechtliche Grundlagen* für Details an.

Beispiel:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Siehe auch:

PRIVACY_URL

2.16.52 LICENSE_EXTRA

Zusätzliche Lizenzen, die in die Lizenzauswahl einbezogen werden sollen.

Bemerkung: Jede Lizenzdefinition sollte ein Tupel aus einem Kurznamen, einem Langnamen und einer URL sein.

Zum Beispiel:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
] ,
```

2.16.53 LICENSE_FILTER

Geändert in Version 4.3: Wenn Sie diesen Wert auf einen leeren Wert setzen, wird die Lizenzwarnung deaktiviert.

Liste der anzuzeigenden Lizenzen filtern. Dadurch wird auch die Lizenzwarnung deaktiviert, wenn sie auf leer gesetzt ist.

Bemerkung: Dieser Filter verwendet die kurzen Lizenznamen.

Zum Beispiel:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

Folgendes deaktiviert die Lizenzwarnung:

```
LICENSE_FILTER = set ()
```

Siehe auch:

alerts

2.16.54 LICENSE_REQUIRED

Legt fest, ob das Lizenzattribut in *Komponentenkonfiguration* erforderlich ist.

Bemerkung: Diese Funktion ist standardmäßig deaktiviert.

2.16.55 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Ob die Länge einer bestimmten Übersetzung begrenzt werden soll. Die Beschränkung ist die Länge der Ausgangszeichenkette × 10 Zeichen.

Hinweis: Setzen Sie dies auf `False`, um längere Übersetzungen (bis zu 10.000 Zeichen) unabhängig von der Länge der Ausgangszeichenkette zu ermöglichen.

Bemerkung: Der Standardwert ist `True`.

2.16.56 LOCALIZE_CDN_URL und LOCALIZE_CDN_PATH

Diese Einstellungen konfigurieren die *JavaScript-Lokalisierung CDN*-Erweiterung. `LOCALIZE_CDN_URL` definiert die Root-URL, unter der das Lokalisierungs-CDN verfügbar ist und `LOCALIZE_CDN_PATH` definiert den Pfad, unter dem Weblate die erzeugten Dateien speichern soll, die unter der `LOCALIZE_CDN_URL` ausgeliefert werden.

Hinweis: Hosted Weblate verwendet `https://weblate-cdn.com/`.

Siehe auch:

JavaScript-Lokalisierung CDN

2.16.57 LOGIN_REQUIRED_URLS

Eine Liste von URLs, für die eine Anmeldung erforderlich sein soll. (Neben den in Weblate eingebauten Standardregeln).

Hinweis: Damit können Sie eine ganze Installation mit einem Passwort schützen:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Hinweis: Es ist wünschenswert, auch den API-Zugang zu sperren, wie im obigen Beispiel gezeigt.

Siehe auch:

REQUIRE_LOGIN

2.16.58 LOGIN_REQUIRED_URLS_EXCEPTIONS

Liste der Ausnahmen für `LOGIN_REQUIRED_URLS`. Wenn nicht angegeben, können Benutzer auf die Anmelde-seite zugreifen.

Einige Ausnahmen, die Sie möglicherweise einschließen möchten:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$", # Required for sign in
    r"/static/(.*)$", # Required for development mode
    r"/widgets/(.*)$", # Allowing public access to widgets
    r"/data/(.*)$", # Allowing public access to data exports
    r"/hooks/(.*)$", # Allowing public access to notification hooks
    r"/api/(.*)$", # Allowing access to API
    r"/js/i18n/$", # JavaScript localization
)
```

2.16.59 MATOMO_SITE_ID

ID einer Website in Matomo (ehemals Piwik), die Sie verfolgen möchten.

Bemerkung: Diese Integration unterstützt den Matomo Tag Manager nicht.

Siehe auch:

MATOMO_URL

2.16.60 MATOMO_URL

Vollständige URL (einschließlich nachgestelltem Schrägstrich) einer Matomo (ehemals Piwik)-Installation, mit der Sie die Nutzung von Weblate verfolgen möchten. Weitere Einzelheiten finden Sie unter <<https://matomo.org/>>.

Hinweis: Diese Integration unterstützt den Matomo Tag Manager nicht.

Zum Beispiel:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Siehe auch:

MATOMO_SITE_ID

2.16.61 NEARBY_MESSAGES

Wie viele Zeichenketten um die aktuell übersetzte Zeichenkette herum angezeigt werden sollen. Dies ist nur ein Standardwert, Benutzer können dies in Benutzerprofil anpassen.

2.16.62 DEFAULT_PAGE_LIMIT

Neu in Version 4.7.

Standardanzahl der Elemente, die bei aktivem Seitenumbruch angezeigt werden.

2.16.63 PAGURE_CREDENTIALS

Neu in Version 4.3.2.

Liste der Zugangsdaten für Pagure-Server.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

Siehe auch:*Pagure-Merge-Requests*, *Pagure-API*

2.16.64 PRIVACY_URL

Neu in Version 4.8.1.

URL, unter der Ihre Weblate-Instanz ihre Datenschutzrichtlinie anzeigt.

Hinweis: Nützlich, wenn Sie Ihre juristischen Dokumente außerhalb von Weblate hosten, um sie in Weblate einzubetten. Bitte schauen Sie sich *Rechtliche Grundlagen* für Details an.

Beispiel:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Siehe auch:*LEGAL_URL*

2.16.65 PRIVATE_COMMIT_EMAIL_OPT_IN

Neu in Version 4.15.

Legt fest, ob es sich bei der privaten Commit-E-Mail um eine Opt-In- oder Opt-Out-E-Mail handelt (standardmäßig ist sie Opt-Out).

Siehe auch:*Benutzerprofil*, *PRIVATE_COMMIT_EMAIL_TEMPLATE*

2.16.66 PRIVATE_COMMIT_EMAIL_TEMPLATE

Neu in Version 4.15.

Vorlage zur Erstellung einer privaten E-Mail für einen Benutzer. Der Standardwert ist "{username}@users.noreply.{site_domain}".

Zum Deaktivieren ein leere Zeichenkette einsetzen.

Bemerkung: Die Verwendung anderer Commit-E-Mails ist für Benutzer freiwillig, es sei denn, sie werden durch *PRIVATE_COMMIT_EMAIL_OPT_IN* konfiguriert. Benutzer können Commit-E-Mails im Profil konfigurieren.

2.16.67 PROJECT_BACKUP_KEEP_COUNT

Neu in Version 4.14.

Legt fest, wie viele Backups pro Projekt auf dem Server gespeichert werden. Der Standardwert ist 3.

Siehe auch:*Sicherungen auf Projektebene*

2.16.68 PROJECT_BACKUP_KEEP_DAYS

Neu in Version 4.14.

Legt fest, wie lange die Projekt-Backups auf dem Server aufbewahrt werden sollen. Die Standardeinstellung ist 30 Tage.

Siehe auch:

Sicherungen auf Projektebene

2.16.69 PROJECT_NAME_RESTRICT_RE

Neu in Version 4.15.

Definiert einen regulären Ausdruck zur Einschränkung der Projektbenennung. Alle übereinstimmenden Namen werden abgelehnt.

Siehe auch:

Projektname

2.16.70 PROJECT_WEB_RESTRICT_RE

Neu in Version 4.15.

Definiert einen regulären Ausdruck zur Einschränkung von Projekt-Websites. Alle übereinstimmenden URLs werden abgelehnt.

Siehe auch:

Projektseite

2.16.71 RATELIMIT_ATTEMPTS

Neu in Version 3.2.

Maximale Anzahl von Authentifizierungsversuchen, bevor eine Ratenbegrenzung angewendet wird.

Der Standardwert ist 5.

Siehe auch:

Ratenbegrenzung, RATELIMIT_WINDOW, RATELIMIT_LOCKOUT

2.16.72 RATELIMIT_WINDOW

Neu in Version 3.2.

Wie lange die Authentifizierung nach der Ratenbegrenzung akzeptiert wird.

Eine Anzahl von Sekunden, die standardmäßig 300 (5 Minuten) beträgt.

Siehe auch:

Ratenbegrenzung, RATELIMIT_ATTEMPTS, RATELIMIT_LOCKOUT

2.16.73 RATELIMIT_LOCKOUT

Neu in Version 3.2.

Wie lange die Authentifizierung nach der Ratenbegrenzung akzeptiert wird.

Eine Anzahl von Sekunden, die standardmäßig 600 (10 Minuten) beträgt.

Siehe auch:

Ratenbegrenzung, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*

2.16.74 REGISTRATION_ALLOW_BACKENDS

Neu in Version 4.1.

Liste der Authentifizierungs-Backends, von denen aus die Registrierung erlaubt werden soll. Dies schränkt nur neue Registrierungen ein, Benutzer können sich weiterhin über alle konfigurierten Authentifizierungs-Backends authentifizieren und Authentifizierungen hinzufügen.

Es wird empfohlen, *REGISTRATION_OPEN* aktiviert zu lassen, während die Registrierungs-Backends eingeschränkt werden. Andernfalls können sich Nutzer zwar registrieren, aber Weblate zeigt keine Links zur Registrierung in der Benutzeroberfläche an.

Beispiel:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Hinweis: Die Backend-Namen entsprechen den in der URL für die Authentifizierung verwendeten Namen.

Siehe auch:

REGISTRATION_OPEN, *Authentifizierung*

2.16.75 REGISTRATION_CAPTCHA

Ein Wert von entweder `True` oder `False`, der angibt, ob die Registrierung neuer Konten durch CAPTCHA geschützt ist. Diese Einstellung ist optional, und der Standardwert `True` wird angenommen, wenn er nicht angegeben wird.

Wenn diese Option aktiviert ist, wird ein CAPTCHA zu allen Seiten hinzugefügt, auf denen ein Benutzer seine E-Mail-Adresse eingibt:

- Registrierung eines neuen Kontos.
- Passwortwiederherstellung.
- E-Mail zu einem Konto hinzufügen.
- Kontaktformular für nicht angemeldete Benutzer.

2.16.76 REGISTRATION_EMAIL_MATCH

Neu in Version 2.17.

Hier können Sie filtern, welche E-Mail-Adressen sich registrieren können.

Die Standardeinstellung ist `.*`, sodass sich jede beliebige E-Mail-Adresse registrieren kann.

Sie können damit die Registrierung auf eine einzige E-Mail-Domäne beschränken:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.77 REGISTRATION_OPEN

Ob die Registrierung von neuen Konten derzeit erlaubt ist. Diese optionale Einstellung kann den Standardwert `True` beibehalten oder in `False` geändert werden.

Diese Einstellung wirkt sich auf die integrierte Authentifizierung per E-Mail-Adresse oder über Python Social Auth aus (Sie können bestimmte Backends mit `REGISTRATION_ALLOW_BACKENDS` auf eine Positivliste setzen).

Bemerkung: Bei der Verwendung von Authentifizierungsmethoden von Drittanbietern, wie z. B. *LDAP-Authentifizierung*, wird lediglich das Registrierungsformular ausgeblendet, aber neue Benutzer können sich möglicherweise trotzdem anmelden und Konten erstellen.

Siehe auch:

REGISTRATION_ALLOW_BACKENDS, REGISTRATION_EMAIL_MATCH, Authentifizierung

2.16.78 REGISTRATION_REBIND

Neu in Version 4.16.

Erlaubt das erneute Binden von Authentifizierungs-Backends für bestehende Benutzer. Aktivieren Sie dies, wenn Sie zwischen Authentifizierungsanbietern migrieren.

Bemerkung: Standardmäßig deaktiviert, um das Hinzufügen anderer Authentifizierungs-Backends zu einem bestehenden Konto zu verhindern. Eine erneute Bindung kann zur Kompromittierung des Kontos führen, wenn mehr Authentifizierungs-Backends von Drittanbietern verwendet werden.

2.16.79 REPOSITORY_ALERT_THRESHOLD

Neu in Version 4.0.2.

Schwellenwert für die Auslösung eines Alarms bei veralteten Repositorys oder Repositorys, die zu viele Änderungen enthalten. Der Standardwert ist 25.

Siehe auch:

`alerts`

2.16.80 REQUIRE_LOGIN

Neu in Version 4.1.

Dies aktiviert `LOGIN_REQUIRED_URLS` und konfiguriert das REST-Framework so, dass eine Authentifizierung für alle API-Endpunkte erforderlich ist.

Bemerkung: Dies ist in der *Sample configuration* implementiert. Für Docker, verwenden Sie `WEBLATE_REQUIRE_LOGIN`.

2.16.81 SENTRY_DSN

Neu in Version 3.9.

Sentry-DSN zur Verwendung für das *Sammeln von Fehlerberichten*.

Siehe auch:

Django-Integration für Sentry

2.16.82 SESSION_COOKIE_AGE_AUTHENTICATED

Neu in Version 4.3.

Setzt den Sitzungsablauf für authentifizierte Benutzer. Dies ergänzt `SESSION_COOKIE_AGE`, das für nicht authentifizierte Benutzer verwendet wird.

Siehe auch:

`SESSION_COOKIE_AGE`

2.16.83 SIMPLIFY_LANGUAGES

Verwenden Sie einfache Sprachcodes für Standard-Sprach/Länder-Kombinationen. Zum Beispiel wird eine `fr_FR`-Übersetzung den `fr` Sprachcode verwenden. Dies ist normalerweise das gewünschte Verhalten, da es die Auflistung von Sprachen für diese Standardkombinationen vereinfacht.

Deaktivieren Sie diese Option, wenn Sie für jede Variante eine andere Übersetzung wünschen.

2.16.84 SITE_DOMAIN

Konfiguriert die Site-Domäne. Dies ist notwendig, um in vielen Bereichen (z. B. Aktivierungs-E-Mails, Benachrichtigungen oder RSS-Feeds) korrekte absolute Links zu erzeugen.

Falls Weblate auf einem Nicht-Standard-Port läuft, geben Sie ihn hier ebenfalls an.

Beispiele:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Bemerkung: Diese Einstellung sollte nur den Domännennamen enthalten. Für die Konfiguration des Protokolls (Aktivierung und Erzwingung von HTTPS) verwenden Sie `ENABLE_HTTPS` und zum Ändern der URL verwenden Sie `URL_PREFIX`.

Hinweis: In einem Docker-Container wird die Site-Domäne über `WEBLATE_ALLOWED_HOSTS` konfiguriert.

Siehe auch:

Seitendomain richtig einstellen, Einrichtung zulässiger Hosts, HTTPS richtig konfigurieren `WEBLATE_SITE_DOMAIN`, `ENABLE_HTTPS`

2.16.85 SITE_TITLE

Seitentitel, der für die Website und gesendete E-Mails verwendet werden soll.

2.16.86 SPECIAL_CHARS

Zusätzliche Zeichen, die in die visuelle Tastatur aufgenommen werden sollen, *Visuelle Tastatur*.

Der Standardwert ist:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "...")
```

2.16.87 SINGLE_PROJECT

Neu in Version 3.8.

Leitet Benutzer direkt zu einem Projekt oder einer Komponente um, anstatt die Übersichtsseite anzuzeigen. Sie können es entweder auf `True` setzen und in diesem Fall funktioniert es nur, wenn es tatsächlich nur ein einziges Projekt in Weblate gibt. Stattdessen können Sie den Projekt-Slug setzen, dann wird die Umleitung bedingungslos zu diesem Projekt erfolgen.

Geändert in Version 3.11: Die Einstellung akzeptiert jetzt auch einen Projekt-Slug, um die Anzeige des einzelnen Projekts zu erzwingen.

Beispiel:

```
SINGLE_PROJECT = "test"
```

2.16.88 SSH_EXTRA_ARGS

Neu in Version 4.9.

Ermöglicht das Hinzufügen benutzerdefinierter Parameter, wenn Weblate SSH aufruft. Dies ist nützlich, wenn Sie sich mit Servern verbinden, die eine herkömmliche Verschlüsselung oder andere nicht standardisierte Funktionen verwenden.

Wenn zum Beispiel die SSH-Verbindung in Weblate mit der Meldung „Unable to negotiate with legacyhost: no matching key exchange method found. Their offer: diffie-hellman-group1-sha1“, können Sie dies aktivieren mit:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"
```

Hinweis: Die Zeichenkette wird von der Shell ausgewertet. Stellen Sie daher sicher, dass alle Leer- und Sonderzeichen in Anführungszeichen gesetzt werden.

Siehe auch:

[OpenSSH-Legacy-Optionen](#)

2.16.89 STATUS_URL

Die URL, unter der Ihre Weblate-Instanz ihren Status meldet.

2.16.90 SUGGESTION_CLEANUP_DAYS

Neu in Version 3.2.1.

Löscht Vorschläge automatisch nach einer bestimmten Anzahl von Tagen. Der Standardwert ist `None`, was bedeutet, dass keine Löschungen vorgenommen werden.

2.16.91 UPDATE_LANGUAGES

Neu in Version 4.3.2.

Steuert, ob die Sprachdatenbank bei der Datenbankmigration aktualisiert werden soll und ist standardmäßig aktiviert. Diese Einstellung hat keinen Einfluss auf den Aufruf von `setuplang`.

Warnung: Die Anzeige der Sprachen kann dadurch inkonsistent werden. Die Sprachdefinitionen von Weblate werden im Laufe der Zeit erweitert und es werden keine Sprachcodes für die definierten Sprachen angezeigt.

Siehe auch:

[Integrierte Sprachdefinitionen](#)

2.16.92 URL_PREFIX

Mit dieser Einstellung können Sie Weblate unter einem bestimmten Pfad ausführen (andernfalls muss es vom Webserver-Root aus gestartet werden).

Bemerkung: Um diese Einstellung zu verwenden, müssen Sie auch Ihren Server so konfigurieren, dass er dieses Präfix entfernt. Bei WSGI kann dies zum Beispiel durch die Einstellung `WSGIScriptAlias` erreicht werden.

Hinweis: Das Präfix sollte mit einem `/` beginnen.

Beispiel:

```
URL_PREFIX = "/translations"
```

Bemerkung: Diese Einstellung funktioniert nicht mit dem in Django eingebauten Server, Sie müssen `urls.py` so anpassen, dass es dieses Präfix enthält.

2.16.93 VCS_API_DELAY

Neu in Version 4.15.1.

Konfiguriert die minimale Verzögerung in Sekunden zwischen API-Aufrufen von Drittanbietern in *GitHub-Pull-Requests*, *GitLab-Merge-Requests*, *Gitea-Pull-Requests*, und *Pagure-Merge-Requests*.

Dadurch werden die API-Aufrufe von Weblate an diese Dienste begrenzt, um sie nicht zu überlasten.

Wenn Sie durch den sekundären Ratenbegrenzer bei GitHub eingeschränkt sind, könnte eine Erhöhung dieses Wertes helfen.

Der Standardwert ist 10.

2.16.94 VCS_BACKENDS

Konfiguration der verfügbaren VCS-Backends.

Bemerkung: Weblate versucht, alle unterstützten Back-Ends zu nutzen, für die Sie Werkzeuge haben.

Hinweis: Sie können damit die Auswahl einschränken oder eigene VCS-Backends hinzufügen.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Siehe auch:

Integration der Versionsverwaltung

2.16.95 VCS_CLONE_DEPTH

Neu in Version 3.10.2.

Legt fest, wie tief das Klonen von Repositorys in Weblate gehen soll.

Bemerkung: Derzeit wird dies nur in *Git* unterstützt. Standardmäßig führt Weblate flache Klone der Repositorys durch, um das Klonen zu beschleunigen und Speicherplatz zu sparen. Je nach Nutzung (z. B. bei der Verwendung von benutzerdefinierten *Erweiterungen*) kann es sinnvoll sein, die Tiefe zu erhöhen oder flache Klone komplett abzuschalten, indem man den Wert 0 setzt.

Hinweis: Falls Sie beim Pushen von Weblate die Fehlermeldung `fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>'` erhalten, schalten Sie flache Klone komplett aus, durch Einstellung von:

```
VCS_CLONE_DEPTH = 0
```

2.16.96 WEBLATE_ADDONS

Liste der zur Verwendung verfügbaren Erweiterungen. Um sie zu verwenden, müssen sie für eine bestimmte Übersetzungskomponente aktiviert werden. Standardmäßig sind alle integrierten Erweiterungen enthalten. Wenn Sie die Liste erweitern, werden Sie wahrscheinlich bestehende Erweiterungen aktiviert lassen wollen, z. B.:

```
WEBLATE_ADDONS = (
    # Built-in add-ons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.LanguangeConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.xml.XMLCustomizeAddon",
    "weblate.addons.properties.PropertiesSortAddon",
    "weblate.addons.git.GitSquashAddon",
    "weblate.addons.removal.RemoveComments",
    "weblate.addons.removal.RemoveSuggestions",
    "weblate.addons.resx.ResxUpdateAddon",
    "weblate.addons.autotranslate.AutoTranslateAddon",
    "weblate.addons.yaml.YAMLCustomizeAddon",
    "weblate.addons.cdn.CDNJSAddon",
    # Add-on you want to include
    "weblate.addons.example.ExampleAddon",
)
```

Bemerkung: Wenn Sie die Erweiterung aus der Liste entfernen, wird es nicht aus den Komponenten deinstalliert. Weblate wird in diesem Fall abstürzen. Bitte deinstallieren Sie die Erweiterung aus allen Komponenten, bevor Sie es aus dieser Liste entfernen.

Siehe auch:

Erweiterungen, DEFAULT_ADDONS

2.16.97 WEBLATE_EXPORTERS

Neu in Version 4.2.

Liste der verfügbaren Exporteure, die das Herunterladen von Übersetzungen oder Glossaren in verschiedenen Dateiformaten anbieten.

Siehe auch:

Unterstützte Dateiformate

2.16.98 WEBLATE_FORMATS

Neu in Version 3.0.

Liste der Dateiformate, die verwendet werden können.

Bemerkung: Die Standardliste enthält bereits die gängigen Formate.

Siehe auch:

Unterstützte Dateiformate

2.16.99 WEBLATE_MACHINERY

Neu in Version 4.13.

Liste der zur Verfügung stehenden Maschinendienste.

Siehe auch:

Automatische Vorschläge konfigurieren

2.16.100 WEBLATE_GPG_IDENTITY

Neu in Version 3.1.

Identität, die von Weblate zum Signieren von Git-Commits verwendet wird, zum Beispiel:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

Der Weblate-GPG-Keyring wird nach einem passenden Schlüssel durchsucht (home/.gnupg unter *DATA_DIR*). Wenn er nicht gefunden wird, wird ein Schlüssel generiert, siehe *Signing Git commits with GnuPG* für weitere Details.

Siehe auch:

Signing Git commits with GnuPG

2.16.101 WEBSITE_REQUIRED

Legt fest, ob *Projektseite* beim Erstellen eines Projekts angegeben werden muss. Standardmäßig aktiviert, da dies für öffentliche Serverkonfigurationen geeignet ist.

2.17 Sample configuration

The following example is shipped as `weblate/settings_example.py` with Weblate:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

import os
import platform
from logging.handlers import SysLogHandler

# Title of site to use
SITE_TITLE = "Weblate"
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # Uncomment for MySQL older than 5.7:
            # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
            # Set emoji capable charset for MySQL:
            # "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away error:
            # "connect_timeout": 28800,
        },
        # Persistent connections
        "CONN_MAX_AGE": 0,
        # Disable server-side cursors, might be needed with pgbouncer
        "DISABLE_SERVER_SIDE_CURSORS": False,
    }
}

# Data directory, you can use following for the development purposes:
# os.path.join(os.path.dirname(os.path.dirname(os.path.abspath(__file__))), "data")
DATA_DIR = "/home/weblate/data"
CACHE_DIR = f"{DATA_DIR}/cache"

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "ไทย"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "简体中文"),
    ("zh-hant", "繁體中文"),
)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(DATA_DIR, "static")

# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# You can generate it using weblate-generate-secret-key
SECRET_KEY = ""

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
        },
    },
]

```

(Fortsetzung auf der nächsten Seite)

```

    },
    "APP_DIRS": True,
}
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_CREDENTIALS = {}

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_CREDENTIALS = {}

# Bitbucket username and token for sending merge requests.
# Please see the documentation for more details.
BITBUCKETSERVER_CREDENTIALS = {}

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth2",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_GITHUB_ORG_KEY = ""
SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""

SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (

```


(Fortsetzung der vorherigen Seite)

```

    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/#account
↪"
SOCIAL_AUTH_PROTECTED_USER_FIELDS = ("email",)
SOCIAL_AUTH_SLUGIFY_USERNAMES = True
SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

# Password validation configuration
AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator" # noqa: E501, pylint: disable=line-too-long
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
]

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"},
    # Optional password strength validation by django-zxcvbn-password
    # {
    #     "NAME": "zxcvbn_password.ZXCVBNValidator",
    #     "OPTIONS": {
    #         "min_score": 3,
    #         "user_attributes": ("username", "email", "full_name")
    #     }
    # },
]

# Password hashing (prefer Argon)
PASSWORD_HASHERS = [
    "django.contrib.auth.hashers.Argon2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher",
    "django.contrib.auth.hashers.BCryptSHA256PasswordHasher",
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
    "weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # Standard Django modules
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.admin.apps.SimpleAdminConfig",
    "django.contrib.admindocs",
    "django.contrib.sitemaps",
    "django.contrib.humanize",
    # Third party Django modules
    "social_django",
    "crispy_forms",
    "crispy_bootstrap3",
    "compressor",
    "rest_framework",
    "rest_framework.authtoken",
    "django_filters",
    "django_celery_beat",
    "corsheaders",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↳ WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - you can also choose "logfile" to log into separate file
#   after configuring it below

# Detect if we can connect to syslog
HAVE_SYSLOG = False
if platform.system() != "Windows":
    try:
        handler = SysLogHandler(address="/dev/log", facility=SysLogHandler.LOG_
↳ LOCAL2)
        handler.close()
        HAVE_SYSLOG = True
    except OSError:
        HAVE_SYSLOG = False

DEFAULT_LOG = "console" if DEBUG or not HAVE_SYSLOG else "syslog"
DEFAULT_LOGLEVEL = "DEBUG" if DEBUG else "INFO"

# A sample logging configuration. The only tangible logging
# performed by this configuration is to send an email to
# the site admins on every HTTP 500 error when DEBUG=False.
# See http://docs.djangoproject.com/en/stable/topics/logging for

```

(Fortsetzung auf der nächsten Seite)

```

# more details on how to customize your logging configuration.
LOGGING = {
    "version": 1,
    "disable_existing_loggers": True,
    "filters": {"require_debug_false": {"()": "django.utils.log.RequireDebugFalse"}
    ↪},
    "formatters": {
        "syslog": {"format": "weblate[%(process)d]: %(levelname)s %(message)s"},
        "simple": {"format": "[% (asctime)s: %(levelname)s/% (process)s] %(message)s
    ↪"},
        "logfile": {"format": "% (asctime)s %(levelname)s %(message)s"},
        "django.server": {
            "()": "django.utils.log.ServerFormatter",
            "format": "[% (server_time)s] %(message)s",
        },
    },
    "handlers": {
        "mail_admins": {
            "level": "ERROR",
            "filters": ["require_debug_false"],
            "class": "django.utils.log.AdminEmailHandler",
            "include_html": True,
        },
        "console": {
            "level": "DEBUG",
            "class": "logging.StreamHandler",
            "formatter": "simple",
        },
        "django.server": {
            "level": "INFO",
            "class": "logging.StreamHandler",
            "formatter": "django.server",
        },
        "syslog": {
            "level": "DEBUG",
            "class": "logging.handlers.SysLogHandler",
            "formatter": "syslog",
            "address": "/dev/log",
            "facility": SysLogHandler.LOG_LOCAL2,
        },
        # Logging to a file
        # "logfile": {
        #     "level": "DEBUG",
        #     "class": "logging.handlers.RotatingFileHandler",
        #     "filename": "/var/log/weblate/weblate.log",
        #     "maxBytes": 100000,
        #     "backupCount": 3,
        #     "formatter": "logfile",
        # },
    },
    "loggers": {
        "django.request": {
            "handlers": ["mail_admins", DEFAULT_LOG],
            "level": "ERROR",
            "propagate": True,
        },
        "django.server": {
            "handlers": ["django.server"],
            "level": "INFO",
            "propagate": False,
        },
    },
}

```

(Fortsetzung der vorherigen Seite)

```

    # Logging database queries
    # "django.db.backends": {
    #     "handlers": [DEFAULT_LOG],
    #     "level": "DEBUG",
    # },
    "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Logging VCS operations
    "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Python Social Auth
    "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Django Authentication Using LDAP
    "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # SAML IdP
    "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    # "weblate.machinery.apertium.ApertiumAPYTranslation",
    # "weblate.machinery.baidu.BaiduTranslation",
    # "weblate.machinery.deepl.DeepLTranslation",
    # "weblate.machinery.glosbe.GlosbeTranslation",
    # "weblate.machinery.google.GoogleTranslation",
    # "weblate.machinery.googlev3.GoogleV3Translation",
    # "weblate.machinery.libretranslate.LibreTranslateTranslation",
    # "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",
    # "weblate.machinery.netease.NeteaseSightTranslation",
    # "weblate.machinery.tmserver.AmagamaTranslation",
    # "weblate.machinery.tmserver.TMServerTranslation",
    # "weblate.machinery.yandex.YandexTranslation",
    # "weblate.machinery.saptranslationhub.SAPTranslationHub",
    # "weblate.machinery.youdao.YoudaoTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APY server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_BAIDU_ID = None
MT_BAIDU_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.
# ↪html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

SECURE_SSL_HOST = SITE_DOMAIN
# Sent referrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",) # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000
# Allow more fields for case with a lot of subscriptions in profile
DATA_UPLOAD_MAX_NUMBER_FIELDS = 2000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer

```

(Fortsetzung auf der nächsten Seite)

```

# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_ALLOWED_TEMPLATE_PACKS = "bootstrap3"
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PerlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.ObjectPascalFormatCheck",
#     "weblate.checks.format.SchemeFormatCheck",
#     "weblate.checks.format.CSharpFormatCheck",
#     "weblate.checks.format.JavaFormatCheck",
#     "weblate.checks.format.JavaMessageFormatCheck",
#     "weblate.checks.format.PercentPlaceholdersCheck",
#     "weblate.checks.format.VueFormattingCheck",
#     "weblate.checks.format.I18NextInterpolationCheck",
#     "weblate.checks.format.ESTemplateLiteralsCheck",
#     "weblate.checks.angularjs.AngularJSInterpolationCheck",
#     "weblate.checks.icu.ICUMessageFormatCheck",
#     "weblate.checks.icu.ICUSourceCheck",
#     "weblate.checks.qt.QtFormatCheck",
#     "weblate.checks.qt.QtPluralCheck",
#     "weblate.checks.ruby.RubyFormatCheck",
#     "weblate.checks.consistency.PluralsCheck",
#     "weblate.checks.consistency.SamePluralsCheck",
#     "weblate.checks.consistency.ConsistencyCheck",
#     "weblate.checks.consistency.TranslatedCheck",
#     "weblate.checks.chars.EscapedNewlineCountingCheck",
#     "weblate.checks.chars.NewLineCountCheck",
#     "weblate.checks.markup.BBCodeCheck",
#     "weblate.checks.chars.ZeroWidthSpaceCheck",
#     "weblate.checks.render.MaxSizeCheck",
#     "weblate.checks.markup.XMLValidityCheck",
#     "weblate.checks.markup.XMLTagsCheck",
#     "weblate.checks.markup.MarkdownRefLinkCheck",

```


(Fortsetzung der vorherigen Seite)

```

# "weblate.checks.markup.MarkdownLinkCheck",
# "weblate.checks.markup.MarkdownSyntaxCheck",
# "weblate.checks.markup.URLCheck",
# "weblate.checks.markup.SafeHTMLCheck",
# "weblate.checks.placeholders.PlaceholderCheck",
# "weblate.checks.placeholders.RegexCheck",
# "weblate.checks.duplicate.DuplicateCheck",
# "weblate.checks.source.OptionalPluralCheck",
# "weblate.checks.source.EllipsisCheck",
# "weblate.checks.source.MultipleFailingCheck",
# "weblate.checks.source.LongUntranslatedCheck",
# "weblate.checks.format.MultipleUnnamedFormatsCheck",
# "weblate.checks.glossary.GlossaryCheck",
# )

# List of automatic fixups
# AUTOFIX_LIST = (
#     "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
#     "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
#     "weblate.trans.autofixes.chars.RemoveZeroSpace",
#     "weblate.trans.autofixes.chars.RemoveControlChars",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LanguaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",
#     "weblate.addons.generate.GenerateFileAddon",
#     "weblate.addons.generate.PseudolocaleAddon",
#     "weblate.addons.generate.PrefillAddon",
#     "weblate.addons.json.JSONCustomizeAddon",
#     "weblate.addons.xml.XMLCustomizeAddon",
#     "weblate.addons.properties.PropertiesSortAddon",
#     "weblate.addons.git.GitSquashAddon",
#     "weblate.addons.removal.RemoveComments",
#     "weblate.addons.removal.RemoveSuggestions",
#     "weblate.addons.resx.ResxUpdateAddon",
#     "weblate.addons.yaml.YAMLCustomizeAddon",
#     "weblate.addons.cdn.CDNJSAddon",
# )

# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

```

(Fortsetzung auf der nächsten Seite)

```

# List of URLs your site is supposed to serve
ALLOWED_HOSTS = ["*"]

# Configuration for caching
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # "LOCATION": "unix:///var/run/redis/redis.sock?db=1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
            # If you set password here, adjust CELERY_BROKER_URL as well
            "PASSWORD": None,
            "CONNECTION_POOL_KWARGS": {},
        },
        "KEY_PREFIX": "weblate",
        "TIMEOUT": 3600,
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
}

# Store sessions in cache
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# Store messages in session
MESSAGE_STORAGE = "django.contrib.messages.storage.session.SessionStorage"

# REST framework settings for API
REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth` permissions,
    # or allow read-only access for unauthenticated users.
    "DEFAULT_PERMISSION_CLASSES": [
        # Require authentication for login required sites
        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "rest_framework.authentication.SessionAuthentication",
    ),
    "DEFAULT_THROTTLE_CLASSES": (
        "weblate.api.throttling.UserRateThrottle",
        "weblate.api.throttling.AnonRateThrottle",
    ),
    "DEFAULT_THROTTLE_RATES": {"anon": "100/day", "user": "5000/hour"},
    "DEFAULT_PAGINATION_CLASS": "weblate.api.pagination.StandardPagination",
    "PAGE_SIZE": 50,
    "VIEW_DESCRIPTION_FUNCTION": "weblate.api.views.get_view_description",
    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL

```

(Fortsetzung der vorherigen Seite)

```

FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = [
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": True},
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": False},
]

# Require login for all URLs
if REQUIRE_LOGIN:
    LOGIN_REQUIRED_URLS = (r"/(.*)$",)

# In such case you will want to include some of the exceptions
# LOGIN_REQUIRED_URLS_EXCEPTIONS = (
#     rf"{URL_PREFIX}/accounts/(.*)$", # Required for login
#     rf"{URL_PREFIX}/admin/login/(.*)$", # Required for admin login
#     rf"{URL_PREFIX}/static/(.*)$", # Required for development mode
#     rf"{URL_PREFIX}/widgets/(.*)$", # Allowing public access to widgets
#     rf"{URL_PREFIX}/data/(.*)$", # Allowing public access to data exports
#     rf"{URL_PREFIX}/hooks/(.*)$", # Allowing public access to notification hooks
#     rf"{URL_PREFIX}/healthz/$", # Allowing public access to health check
#     rf"{URL_PREFIX}/api/(.*)$", # Allowing access to API
#     rf"{URL_PREFIX}/js/i18n/$", # JavaScript localization
#     rf"{URL_PREFIX}/contact/$", # Optional for contact form
#     rf"{URL_PREFIX}/legal/(.*)$", # Optional for legal app
#     rf"{URL_PREFIX}/avatar/(.*)$", # Optional for avatars
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True
# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULER = "django_celery_beat.schedulers:DatabaseScheduler"
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate*": {"queue": "translate"},
    "weblate.accounts.tasks.notify*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# CORS allowed origins
CORS_ALLOWED_ORIGINS = []
CORS_URLS_REGEX = r"^/api/.*$"

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

# PGP commits signing
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None
SENTRY_DSN = None
SENTRY_ENVIRONMENT = SITE_DOMAIN
AKISMET_API_KEY = None
```

2.18 Management commands

Bemerkung: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check [Dateisystemberechtigungen](#) for more details.

You will find basic management commands (available as `./manage.py` in the Django sources, or as an extended set in a script called **weblate** installable atop Weblate).

2.18.1 Invoking management commands

As mentioned before, invocation depends on how you installed Weblate.

If using virtualenv for Weblate, you can either specify the full path to **weblate**, or activate the virtualenv prior to invoking it:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

If you are using source code directly (either from a tarball or Git checkout), the management script is `./manage.py` available in the Weblate sources. To run it:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip installer, or by using the `./setup.py` script, the **weblate** is installed to your path (or virtualenv path), from where you can use it to control Weblate:

```
weblate list_versions
```

For the Docker image, the script is installed like above, and you can run it using **docker exec**:

```
docker exec --user weblate <container> weblate list_versions
```

For **docker-compose** the process is similar, you just have to use **docker-compose exec**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

In case you need to pass it a file, you can temporary add a volume:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/
↪users.json
```

Siehe auch:

Installation über Docker, Installation unter Debian und Ubuntu, Installation unter SUSE und openSUSE, Installation unter RedHat, Fedora und CentOS, Installieren aus Quellen

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Neu in Version 2.5.

Imports a translation from the file to use as a suggestion for the given translation. It skips duplicated translations; only different ones are added.

--author USER@EXAMPLE.COM

E-mail of author for the suggestions. This user has to exist prior to importing (you can create one in the admin interface if needed).

Beispiel:

```
weblate --author michal@cihar.com add_suggestions weblate application cs /tmp/
↪suggestions-cs.po
```

2.18.3 auto_translate

weblate auto_translate <project> <component> <language>

Neu in Version 2.5.

Geändert in Version 4.6: Parameter für Übersetzungsmodus hinzugefügt.

Performs automatic translation based on other component translations.

--source PROJECT/COMPONENT

Specifies the component to use as source available for translation. If not specified all components in the project are used.

--user USERNAME

Specify username listed as author of the translations. „Anonymous user“ is used if not specified.

--overwrite

Whether to overwrite existing translations.

--inconsistent

Whether to overwrite existing translations that are inconsistent (see *Inkonsistent*).

--add

Automatically add language if a given translation does not exist.

--mt MT

Use machine translation instead of other components as machine translations.

--threshold THRESHOLD

Similarity threshold for machine translation, defaults to 80.

--mode MODE

Geben Sie den Übersetzungsmodus an, Standard ist `translate`, aber `fuzzy` oder `suggest` können verwendet werden.

Beispiel:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application_
↪weblate website cs
```

Siehe auch:

Automatische Übersetzung

2.18.4 celery_queues

weblate celery_queues

Neu in Version 3.7.

Displays length of Celery task queues.

2.18.5 checkgit

weblate checkgit <project|project/component>

Prints current state of the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

2.18.6 commitgit

weblate commitgit <project|project/component>

Commits any possible pending changes to the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components, or use `--file-format` to filter based on the file format.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Commits pending changes older than a given age.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

--age HOURS

Age in hours for committing. If not specified the value configured in *Komponentenkonfiguration* is used.

Bemerkung: This is automatically performed in the background by Weblate, so there no real need to invoke this manually, besides forcing an earlier commit than specified by *Komponentenkonfiguration*.

Siehe auch:

Ausführen von Wartungsaufgaben, COMMIT_PENDING_HOURS

2.18.8 cleanuptrans

weblate cleanuptrans

Cleans up orphaned checks and translation suggestions. There is normally no need to run this manually, as the cleanups happen automatically in the background.

Siehe auch:

Ausführen von Wartungsaufgaben

2.18.9 cleanup_ssh_keys

weblate cleanup_ssh_keys

Neu in Version 4.9.1.

Performs cleanup of stored SSH host keys:

- Removes deprecated RSA keys for GitHub which might cause issues connecting to GitHub.
- Removes duplicate entries in host keys.

Siehe auch:

SSH-Repositoryys

2.18.10 createadmin

weblate createadmin

Creates an `admin` account with a random password, unless it is specified.

--password PASSWORD

Provides a password on the command-line, to not generate a random one.

--no-password

Do not set password, this can be useful with *-update*.

--username USERNAME

Use the given name instead of `admin`.

--email USER@EXAMPLE.COM

Specify the admin e-mail address.

--name

Specify the admin name (visible).

--update

Update the existing user (you can use this to change passwords).

Geändert in Version 2.9: Added parameters `--username`, `--email`, `--name` and `--update`.

2.18.11 dump_memory

weblate dump_memory

Neu in Version 2.20.

Exportieren Sie eine JSON-Datei mit Weblate-Übersetzungsspeicher-Inhalten.

Siehe auch:

Übersetzungsspeicher, *Weblate Übersetzungsspeicher-Schema*

2.18.12 dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by *importuserdata*.

Hinweis: This comes in handy when migrating or merging Weblate instances.

2.18.13 import_demo

weblate import_demo

Neu in Version 4.1.

Creates a demo project with components based on <https://github.com/WeblateOrg/demo>. Make sure the celery tasks are running before running this command.

This can be useful when developing Weblate.

2.18.14 import_json

weblate import_json <json-file>

Neu in Version 2.7.

Batch import of components based on JSON data.

The imported JSON file structure pretty much corresponds to the component object (see *GET /api/components/(string:project)/(string:component)/*). You have to include the name and filemask fields.

--project PROJECT

Specifies where the components will be imported from.

--main-component COMPONENT

Use the given VCS repository from this component for all of them.

--ignore

Skip (already) imported components.

--update

Update (already) imported components.

Geändert in Version 2.9: The parameters `--ignore` and `--update` are there to deal with already imported components.

Beispiel für eine JSON-Datei:

```
[
  {
    "slug": "po",
    "name": "Gettext PO",
    "file_format": "po",
    "filemask": "po/*.po",
    "new_lang": "none"
  },
  {
    "name": "Android",
    "filemask": "android/values-*/strings.xml",
    "template": "android/values/strings.xml",
    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]
```

Siehe auch:

import_memory

2.18.15 import_memory

weblate import_memory <file>

Neu in Version 2.20.

Importiert eine TMX- oder JSON-Datei in den Weblate-Übersetzungsspeicher.

--language-map LANGMAP

Ermöglicht die Zuordnung von Sprachen im TMX zum Weblate-Übersetzungsspeicher. Die Sprachcodes werden nach der Normalisierung, die normalerweise von Weblate vorgenommen wird, zugeordnet.

`--language-map en_US:en` will for example import all `en_US` strings as `en` ones.

This can be useful in case your TMX file locales happen not to match what you use in Weblate.

Siehe auch:

Übersetzungsspeicher, Weblate Übersetzungsspeicher-Schema

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

Geändert in Version 3.0: The `import_project` command is now based on the *Komponentenerkennung* add-on, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on the file mask.

`<project>` names an existing project, into which the components are to be imported.

The `<gitrepo>` defines the Git repository URL to use, and `<branch>` signifies the Git branch. To import additional translation components from an existing Weblate component, use a `weblate://<project>/<component>` URL for the `<gitrepo>`.

The `<filemask>` defines file discovery for the repository. It can be either be made simple using wildcards, or it can use the full power of regular expressions.

The simple matching uses `**` for component name and `*` for language, for example: `**/*.po`

The regular expression has to contain groups named *component* and *language*. For example: `(?P<language>[^\s]*) / (?P<component>[^\s]*) \.po`

The import matches existing components based on files and adds the ones that do not exist. It does not change already existing ones.

--name-template TEMPLATE

Customize the name of a component using Django template syntax.

For example: `Documentation: {{ component }}`

--base-file-template TEMPLATE

Customize the base file for monolingual translations.

For example: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Customize the base file for addition of new translations.

For example: `{{ component }}/ts/en.ts`

--file-format FORMAT

You can also specify the file format to use (see *Unterstützte Dateiformate*), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see *Komponentenkonfiguration*) with this parameter. It has to be a valid regular expression.

--main-component

You can specify which component will be chosen as the main one—the one actually containing the VCS repository.

--license NAME

Specify the overall, project or component translation license.

--license-url URL

Specify the URL where the translation license is to be found.

--vcs NAME

In case you need to specify which version control system to use, you can do it here. The default version control is Git.

To give you some examples, let's try importing two projects.

First The Debian Handbook translations, where each language has separate a folder with the translations of each chapter:

```
weblate import_project \
  debian-handbook \
  git://anonscm.debian.org/debian-handbook/debian-handbook.git \
  squeeze/master \
  '*/**.po'
```

Then the Tanaguru tool, where the file format needs be specified, along with the base file template, and how all components and translations are located in single folder:

```
weblate import_project \
  --file-format=properties \
  --base-file-template=web-app/tgol-web-app/src/main/resources/i18n/%s-I18N.
  ↪properties \
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
tanaguru \
https://github.com/Tanaguru/Tanaguru \
master \
web-app/tgol-web-app/src/main/resources/i18n/**-I18N*.properties
```

More complex example of parsing of filenames to get the correct component and language out of a filename like `src/security/Numerous_security_holes_in_0.10.1.de.po`:

```
weblate import_project \
tails \
git://git.tails.boum.org/tails master \
'wiki/src/security/(?P<component>.*).\.(?P<language>[^.]*)\.po$'
```

Filtering only translations in a chosen language:

```
./manage import_project \
--language-regex '^(cs|sk)$' \
weblate \
https://github.com/WeblateOrg/weblate.git \
'weblate/locale/*/LC_MESSAGES/**/*.po'
```

Importing Sphinx documentation split to multiple files:

```
$ weblate import_project --name-template 'Documentation: %s' \
--file-format po \
project https://github.com/project/docs.git master \
'docs/locale/*/LC_MESSAGES/**/*.po'
```

Importing Sphinx documentation split to multiple files and directories:

```
$ weblate import_project --name-template 'Directory 1: %s' \
--file-format po \
project https://github.com/project/docs.git master \
'docs/locale/*/LC_MESSAGES/dir1/**/*.po'
$ weblate import_project --name-template 'Directory 2: %s' \
--file-format po \
project https://github.com/project/docs.git master \
'docs/locale/*/LC_MESSAGES/dir2/**/*.po'
```

Siehe auch:

More detailed examples can be found in the starting chapter, alternatively you might want to use *import_json*.

2.18.17 importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by *dumpuserdata*

2.18.18 importusers

weblate importusers --check <file.json>

Imports users from JSON dump of the Django auth_users database.

--check

With this option it will just check whether a given file can be imported and report possible conflicts arising from usernames or e-mails.

You can dump users from the existing Django installation using:

```
weblate dumpdata auth.User > users.json
```

2.18.19 install_addon

Neu in Version 3.2.

weblate install_addon --addon ADDON <project|project/component>

Installs an add-on to a set of components.

--addon ADDON

Name of the add-on to install. For example `weblate.gettext.customize`.

--configuration CONFIG

JSON-kodierte Konfiguration einer Erweiterung.

--update

Aktualisieren Sie die vorhandene Konfiguration der Erweiterung.

You can either define which project or component to install the add-on in (for example `weblate/application`), or use `--all` to include all existing components.

To install *Ausgabe von Gettext anpassen* for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --  
↪update --all
```

Siehe auch:

Erweiterungen

2.18.20 list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names.

This is used to generate `<https://wiki.l10n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>`.

2.18.21 list_translators

weblate list_translators <project|project/component>

Lists translators by contributed language for the given project:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code

List names by language code instead of language name.

You can either define which project or component to use (for example `weblate/application`), or use `--all` to list translators from all existing components.

2.18.22 list_versions

weblate list_versions

Lists all Weblate dependencies and their versions.

2.18.23 loadpo

weblate loadpo <project|project/component>

Reloads translations from disk (for example in case you have done some updates in the VCS repository).

--force

Force update, even if the files should be up-to-date.

--lang LANGUAGE

Limit processing to a single language.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Bemerkung: You seldom need to invoke this, Weblate will automatically load changed files for every VCS update. This is needed in case you manually changed an underlying Weblate VCS repository or in some special cases following an upgrade.

2.18.24 lock_translation

weblate lock_translation <project|project/component>

Prevents further translation of a component.

Hinweis: Useful in case you want to do some maintenance on the underlying repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Siehe auch:

`unlock_translation`

2.18.25 move_language

weblate move_language source target

Neu in Version 3.0.

Allows you to merge language content. This is useful when updating to a new version which contains aliases for previously unknown languages that have been created with the (*generated*) suffix. It moves all content from the *source* language to the *target* one.

Beispiel:

```
weblate move_language cze cs
```

After moving the content, you should check whether there is anything left (this is subject to race conditions when somebody updates the repository meanwhile) and remove the (*generated*) language.

2.18.26 pushgit

weblate pushgit <project|project/component>

Pushes committed changes to the upstream VCS repository.

--force-commit

Force commits any pending changes, prior to pushing.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Bemerkung: Weblate pushes changes automatically if *Bei Commit gleichzeitig Pushen* in *Komponentenkonfiguration* is turned on, which is the default.

2.18.27 unlock_translation

weblate unlock_translation <project|project/component>

Unlocks a given component, making it available for translation.

Hinweis: Useful in case you want to do some maintenance on the underlying repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Siehe auch:

lock_translation

2.18.28 setupgroups

weblate setupgroups

Configures default groups and optionally assigns all users to that default group.

--no-privs-update

Turns off automatic updating of existing groups (only adds new ones).

--no-projects-update

Prevents automatic updates of groups for existing projects. This allows adding newly added groups to existing projects, see *Projekt-Zugriffssteuerung*.

Siehe auch:

Liste der Berechtigungen und integrierten Rollen

2.18.29 setuplang

weblate setuplang

Updates list of defined languages in Weblate.

--no-update

Turns off automatic updates of existing languages (only adds new ones).

2.18.30 updatechecks

weblate updatechecks <project|project/component>

Updates all checks for all strings.

Hinweis: Useful for upgrades which do major changes to checks.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

2.18.31 updategit

weblate updategit <project|project/component>

Fetches remote VCS repositories and updates the internal cache.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Bemerkung: Usually it is better to configure hooks in the repository to trigger *Benachrichtigungs-Hooks*, instead of regular polling by *updategit*.

2.19 Ankündigungen

Geändert in Version 4.0: In früheren Versionen wurde diese Funktion Whiteboard-Nachrichten genannt.

Versorgen Sie Ihre Übersetzer mit Informationen, indem Sie Ankündigungen auf der gesamten Website, pro Projekt, Komponente oder Sprache veröffentlichen.

Geben Sie den Zweck, die Fristen, den Status oder die Ziele für die Übersetzung an.

Die Nutzer werden über die Ankündigungen der beobachteten Projekte benachrichtigt (sofern sie sich nicht abmelden).

Dies kann für verschiedene Dinge nützlich sein, von der Ankündigung des Zwecks der Website bis zur Angabe von Zielen für Übersetzungen.

Die Ankündigungen können auf jeder Ebene im Menü *Verwaltung* mit *Ankündigung* veröffentlicht werden:

The screenshot shows the Weblate web interface. At the top, a dark navigation bar contains the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this, a light blue banner displays the message: "Translations will be used only if they reach 60%." The main content area has a top navigation bar with links: Components, Languages, Info, Search, Insights, Files, Tools, Manage (highlighted), and Share. On the right, it says "Not watching". The "Post announcement" form is open, showing a large text area for the message, a dropdown for the category (currently "Info (light blue)"), a date picker for the expiry date (set to "mm/dd/yyyy"), and a checked checkbox for "Notify users". A blue "Add" button is at the bottom of the form.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Sie kann auch über die Adminoberfläche hinzugefügt werden:

Weblate administration
WELCOME, **WEBLATE TEST** / [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Announcements · Add Announcement

Add Announcement

Required fields are marked in bold.

Message:

Translations will be used only if they reach 60%

You can use Markdown and mention users by @username.

Project: WeblateOrg

Component:

Language:

Category: Info (light blue)
Category defines color used for the message.

Expiry date: Today

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ Notify users

Save and add another
Save and continue editing
SAVE

Die Ankündigungen werden dann in dem angegebenen Kontext angezeigt:

Kein Kontext angegeben

Wird auf der Übersicht (Einstiegsseite) angezeigt.

Projekt spezifiziert

Wird im Projekt angezeigt, einschließlich aller Komponenten und Übersetzungen.

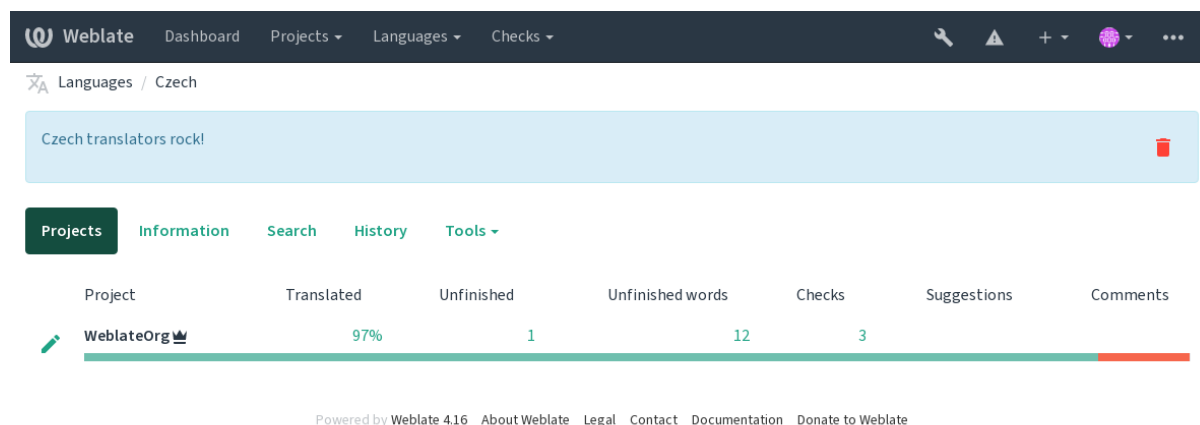
Komponente spezifiziert

Wird für eine bestimmte Komponente und alle ihre Übersetzungen angezeigt.

Sprache spezifiziert

Wird in der Sprachübersicht und allen Übersetzungen in dieser Sprache angezeigt.

So sieht es auf der Übersichtsseite der Sprachen aus:



2.20 Komponentenlisten

Geben Sie mehrere Listen von Komponenten an, die als Optionen auf dem Benutzer-Dashboard erscheinen sollen, aus denen die Benutzer eine als Standardansicht auswählen können. Siehe [Übersichtsseite](#) um mehr zu erfahren.

Geändert in Version 2.20: Für jede auf der Übersichtsseite angezeigte Komponentenliste wird ein Status angezeigt.

Die Namen und Inhalte der Komponentenlisten können in der Verwaltungsoberfläche im Abschnitt *Komponentenlisten* festgelegt werden. Jede Komponentenliste muss einen Namen haben, der dem Benutzer angezeigt wird, und einen Slug, der sie in der URL repräsentiert.

Geändert in Version 2.13: Ändern Sie die Dashboard-Einstellungen für anonyme Benutzer von der Verwaltungsoberfläche aus, indem Sie ändern, welches Dashboard nicht authentifizierten Benutzern angezeigt wird.

2.20.1 Automatische Komponentenlisten

Neu in Version 2.13.

Fügen Sie der Liste automatisch Komponenten auf der Grundlage ihres Slugs hinzu, indem Sie Regeln für die *Automatische Komponentenlistenzuordnung* erstellen.

- Nützlich für die Pflege von Komponentenlisten für große Installationen oder für den Fall, dass Sie eine Komponentenliste mit allen Komponenten Ihrer Weblate-Installation haben möchten.

Hinweis: Erstellen Sie eine Komponentenliste mit allen Komponenten Ihrer Weblate-Installation.

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:

Weblate administration
WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Component lists · Add Component list

Add Component list

Required fields are marked in bold.

Component list name:
Display name

URL slug:
Name used in URLs and filenames.

☒ Show on dashboard
When enabled this component list will be shown as a tab on the dashboard

Components:

Available components ⓘ

- WeblateOrg/Django
- WeblateOrg/Language names
- WeblateOrg/WeblateOrg

Choose all ⓘ

Chosen components ⓘ +
Remove all ⓘ

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE? ⓘ
<input type="text" value="^.*\$"/>	<input type="text" value="^.*\$"/>	<input type="button" value="✕"/>

+ Add another Automatic component list assignment

Save and add another

Save and continue editing

SAVE

2.21 Optionale Weblate-Module

Für Ihre Einrichtung sind mehrere optionale Module verfügbar.

2.21.1 Git-Exporter

Neu in Version 2.10.

Ermöglicht Ihnen den Nur-Lese-Zugriff auf das zugrunde liegende Git-Repository über HTTP(S).

Installation

1. `weblate.gitexport` zu installierten Anwendungen in `settings.py` hinzufügen:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Vorhandene Repositories exportieren, indem Sie Ihre Datenbank nach der Installation migrieren:

```
weblate migrate
```

Anwendung

Das Modul klinkt sich automatisch in Weblate ein und setzt die exportierte Repository-URL in der *Komponenten-konfiguration*. Die Repositories sind unter dem `/git/`-Teil der Weblate-URL zugänglich, zum Beispiel `https://example.org/git/weblate/main/`.

Repositories für öffentlich zugängliche Projekte können ohne Authentifizierung geklont werden:

```
git clone 'https://example.org/git/weblate/main/'
```

Access to browse the repositories with restricted access (with *Private access control* or when `REQUIRE_LOGIN` is enabled) requires an API token which can be obtained in your *user profile*:

```
git clone 'https://user:KEY@example.org/git/weblate/main/'
```

Hinweis: Standardmäßig haben Mitglieder der Gruppe *Benutzer* und anonyme Benutzer über die Rollen *Auf Repository zugreifen* und *Hauptbenutzer Zugriff* auf die Repositories für öffentliche Projekte.

2.21.2 Abrechnung

Neu in Version 2.4.

This is used on [Hosted Weblate](#) to define billing plans, track invoices and usage limits.

Installation

1. Add `weblate.billing` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

2. Run the database migration to optionally install additional database structures for the module:

```
weblate migrate
```

Anwendung

After installation you can control billing in the admin interface. Users with billing enabled will get new *Billing* tab in their *Benutzerprofil*.

The billing module additionally allows project admins to create new projects and components without being superusers (see *Adding translation projects and components*). This is possible when following conditions are met:

- The billing is in its configured limits (any overusage results in blocking of project/component creation) and paid (if its price is non zero)
- The user is admin of existing project with billing or user is owner of billing (the latter is necessary when creating new billing for users to be able to import new projects).

Upon project creation user is able to choose which billing should be charged for the project in case he has access to more of them.

2.21.3 Rechtliche Grundlagen

Neu in Version 2.15.

This is used on *Hosted Weblate* to provide required legal documents. It comes provided with blank documents, and you are expected to fill out the following templates in the documents:

legal/documents/tos.html

Terms of service document

legal/documents/privacy.html

Privacy policy document

legal/documents/summary.html

Short overview of the terms of service and privacy policy

On changing the terms of service documents, please adjust `LEGAL_TOS_DATE` so that users are forced to agree with the updated documents.

Bemerkung: Legal documents for the Hosted Weblate service are available in this Git repository <<https://github.com/WeblateOrg/wllegal/tree/main/wllegal/templates/legal/documents>>.

Most likely these will not be directly usable to you, but might come in handy as a starting point if adjusted to meet your needs.

Installation

1. Add `weblate.legal` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.legal",)

# Optional:

# Social auth pipeline to confirm TOS upon registration/subsequent sign in
SOCIAL_AUTH_PIPELINE += ("weblate.legal.pipeline.tos_confirm",)

# Middleware to enforce TOS confirmation of signed in users
MIDDLEWARE += [
    "weblate.legal.middleware.RequireTOSMiddleware",
]
```

2. Run the database migration to optionally install additional database structures for the module:

```
weblate migrate
```

3. Edit the legal documents in the `weblate/legal/templates/legal/` folder to match your service.

Anwendung

After installation and editing, the legal documents are shown in the Weblate UI.

2.21.4 Avatars

Avatars are downloaded and cached server-side to reduce information leaks to the sites serving them by default. The built-in support for fetching avatars from e-mails addresses configured for it can be turned off using `ENABLE_AVATARS`.

Weblate currently supports:

- Gravatar
- Libravatar

Siehe auch:

Avatar-Caching, `AVATAR_URL_PREFIX`, `ENABLE_AVATARS`

2.21.5 Spam protection

You can protect against spamming by users by using the [Akismet](#) service.

1. Install the *akismet* Python module (this is already included in the official Docker image).
2. Obtain the Akismet API key.
3. Store it as `AKISMET_API_KEY` or `WEBLATE_AKISMET_API_KEY` in Docker.

Following content is sent to Akismet for checking:

- Vorschläge von unautorisierten Benutzern
- Projekt- und Komponentenbeschreibungen und Links

Bemerkung: This (among other things) relies on IP address of the client, please see *Hinter einem Reverse-Proxy ausführen* for properly configuring that.

Siehe auch:

Hinter einem Reverse-Proxy ausführen, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Signing Git commits with GnuPG

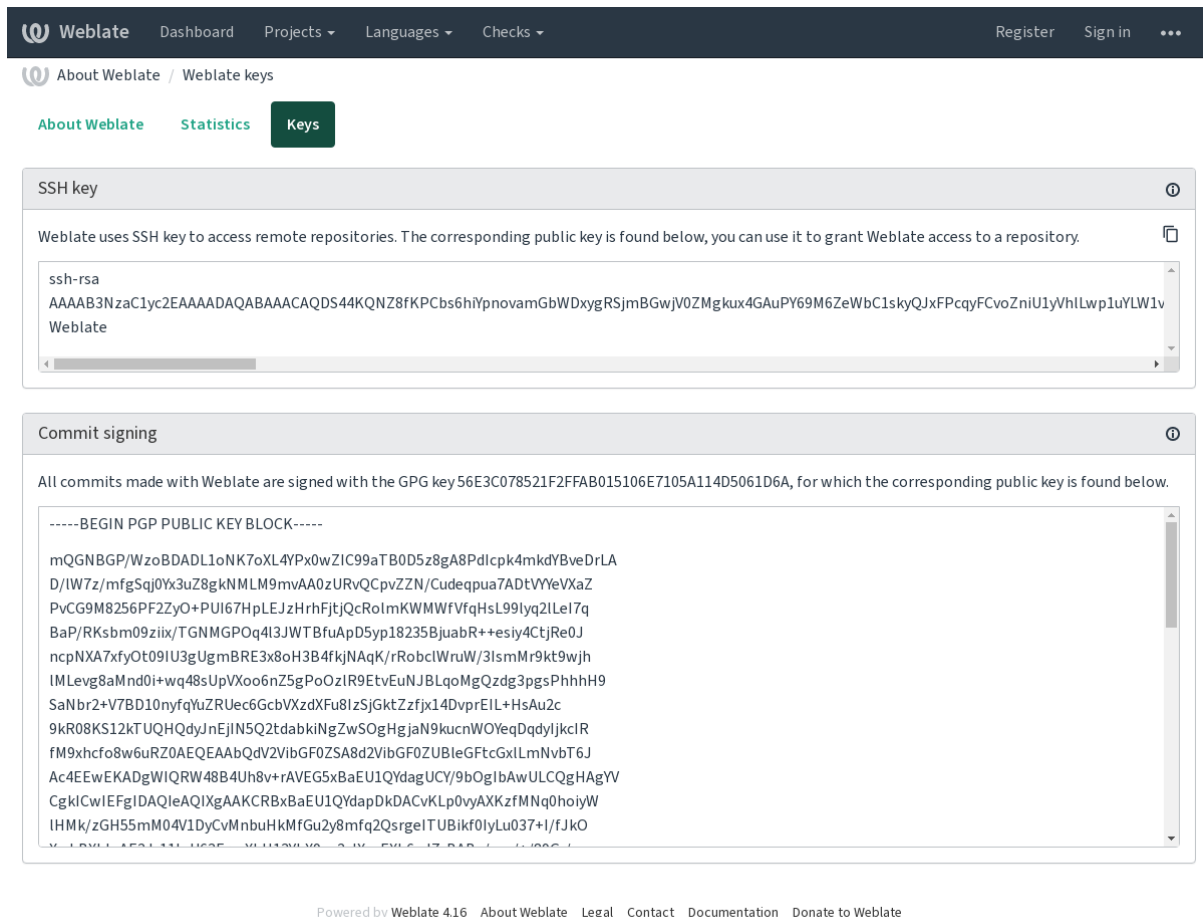
Neu in Version 3.1.

All commits can be signed by the GnuPG key of the Weblate instance.

1. Turn on `WEBLATE_GPG_IDENTITY`. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

This feature needs GnuPG 2.1 or newer installed.

You can find the key in the `DATA_DIR` and the public key is shown on the „About“ page:



SSH key

Weblate uses SSH key to access remote repositories. The corresponding public key is found below, you can use it to grant Weblate access to a repository.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNz8FKPCbs6hiYpnovamGbWdxygRSjmbGwJV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhlLwp1uYlW1v
Weblate
```

Commit signing

All commits made with Weblate are signed with the GPG key 56E3C078521F2FFAB015106E7105A114D5061D6A, for which the corresponding public key is found below.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGP/WzoBDADL1oNK7oXL4YPx0wZIC99aTB0D5z8gA8Pdlcpk4mkdYBveDrLA
D/IW7z/mfgSqj0Yx3uZ8gkNMLM9mvAA0zURvQCpvZZN/Cudeqqua7ADtVYVeXaZ
PvCG9M8256PF2ZyO+PUI67HpLEJzHrhFjtjQcRoImKWMWfVfqHsL99lyq2lLeI7q
BaP/RKsbm09ziix/TGNMGPOq4l3JWTFbuApD5yp18235BjuabR++esiy4CtjRe0J
ncpNXA7xfyOt09IU3gUgmBRE3x8oH3B4fkjNAqK/rRobclWruW/3lsmMr9kt9wjh
lMLevg8aMnd0i+wq48sUpVXoo6nZ5gPoOzIR9EtvEuNJBLoMgQzdg3pgsPhhhH9
SaNbr2+V7BD10nyfqYuZRUec6GcbVXzdXfu8IzSjGktZzfjx14DvprEIL+HsAu2c
9kR08KS12kTUQHQdyJnEjIN5Q2tdabkiNgZwSOGHgjaN9kucnWOYeqDqdyIjklR
fM9xhcf08w6uRZ0AEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWlQRW48B4Uh8v+rAVEG5xBaEU1QYdagUCY/9bOglbAwULCQgHAgYV
CgklCwIEFgIDAQIeAQIXgAAKCRBxBaEU1QYdapDkDACvKLp0vyAXKzfMNq0hoiyW
lHMK/zGH55mM04V1DyCvMnbuHkMfGu2y8mfq2QsrgelTUBikf0lyLu037+I/fJkO
-----
```

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking `gpg`.

Siehe auch:

`WEBLATE_GPG_IDENTITY`

2.21.7 Ratenbegrenzung

Geändert in Version 3.2: Die Ratenbegrenzung lässt nun eine feiner abgestufte Konfiguration zu.

Geändert in Version 4.6: The rate limiting no longer applies to superusers.

Several operations in Weblate are rate limited. At most `RATELIMIT_ATTEMPTS` attempts are allowed within `RATELIMIT_WINDOW` seconds. The user is then blocked for `RATELIMIT_LOCKOUT`. There are also settings specific to scopes, for example `RATELIMIT_CONTACT_ATTEMPTS` or `RATELIMIT_TRANSLATE_ATTEMPTS`. The table below is a full list of available scopes.

The following operations are subject to rate limiting:

Name	Zielgruppe	Allowed tempts	at-	Ratelimit win- dow	Lockout pe- riod
Registrierung	REGISTRA- TION	5		300	600
Nachricht an Administratoren senden	MESSAGE	2		300	600
Passwort-Authentifizierung bei der An- meldung	LOGIN	5		300	600
Plattformweite Suche	SEARCH	6		60	60
Translating	TRANSLATE	30		60	600
Adding to glossary	GLOSSARY	30		60	600
Beginn der Übersetzung in eine neue Sprache	LANGUAGE	2		300	600
Neues Projekt erstellen	PROJECT	5		600	600

If a user fails to sign in `AUTH_LOCK_ATTEMPTS` times, password authentication will be turned off on the account until having gone through the process of having its password reset.

The settings can be also applied in the Docker container by adding `WEBLATE_` prefix to the setting name, for example `RATELIMIT_ATTEMPTS` becomes `WEBLATE_RATELIMIT_ATTEMPTS`.

The API has separate rate limiting settings, see *API-Ratenbegrenzung*.

Siehe auch:

Ratenbegrenzung, Hinter einem Reverse-Proxy ausführen, API-Ratenbegrenzung

2.21.8 Integration von Fedora-Messaging

Fedora Messaging is AMQP-based publisher for all changes happening in Weblate. You can hook additional services on changes happening in Weblate using this.

The Fedora Messaging integration is available as a separate Python module `weblate-fedora-messaging`. Please see <https://github.com/WeblateOrg/fedora_messaging/> for setup instructions.

2.22 Anpassen von Weblate

Erweiterungen und Anpassungen mit Django und Python. Tragen Sie Ihre Änderungen Upstream bei, damit alle davon profitieren können. Das reduziert Ihre Wartungskosten; Code in Weblate wird bei Änderungen an internen Schnittstellen oder beim Refactoring des Codes berücksichtigt.

Warnung: Weder interne Schnittstellen noch Vorlagen gelten als stabile API. Bitte überprüfen Sie bei jedem Upgrade Ihre eigenen Anpassungen, da sich die Schnittstellen oder ihre Semantik ohne Vorankündigung ändern können.

Siehe auch:

Zu Weblate beitragen

2.22.1 Erstellung eines Python-Moduls

Wenn Sie mit Python noch nicht vertraut sind, sollten Sie sich [Python für Anfänger](#) ansehen, wo die Grundlagen erklärt werden und auf weitere Tutorials verwiesen wird.

Um eine Datei mit benutzerdefiniertem Python-Code (Modul genannt) zu schreiben, wird ein Speicherort benötigt, entweder im Systempfad (normalerweise etwas wie `/usr/lib/python3.9/site-packages/`) oder im Weblate-Verzeichnis, das auch dem Suchpfad des Interpreters hinzugefügt wird.

Neu in Version 3.8-5: Wenn Sie [Docker](#) verwenden, können Sie Python-Module in `/app/data/python/` ablegen (siehe [Docker-Container-Volumes](#)), sodass sie von Weblate geladen werden können, zum Beispiel aus einer [Settings-Override-Datei](#).

Noch besser ist es, wenn Sie Ihre Anpassungen in ein richtiges Python-Paket verwandeln:

1. Erstellen Sie einen Ordner für Ihr Paket (wir werden `weblate_customization` verwenden).
2. Erstellen Sie darin eine Datei `setup.py`, um das Paket zu beschreiben:

```
from setuptools import setup

setup(
    name="weblate_customization",
    version="0.0.1",
    author="Your name",
    author_email="yourname@example.com",
    description="Sample Custom check for Weblate.",
    license="GPLv3+",
    keywords="Weblate check example",
    packages=["weblate_customization"],
)
```

3. Erstellen Sie einen Ordner für das Python-Modul (auch `weblate_customization` genannt) für den Anpassungscode.
4. Erstellen Sie darin eine `__init__.py`-Datei, um sicherzustellen, dass Python das Modul importieren kann.
5. Dieses Paket kann nun mit `pip install -e` installiert werden. Weitere Informationen sind in [Editable installs](#) zu finden.
6. Nach der Installation kann das Modul in der Weblate-Konfiguration verwendet werden (zum Beispiel `weblate_customization.checks.FooCheck`).

Die Struktur Ihres Pakets sollte wie folgt aussehen:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

Ein Beispiel für die Anpassung von Weblate finden Sie unter <https://github.com/WeblateOrg/customize-example>, es deckt alle unten beschriebenen Themen ab.

2.22.2 Ändern des Logos

1. Erstellen Sie eine einfache Django-App, die die statischen Dateien enthält, die Sie überschreiben möchten (siehe *Erstellung eines Python-Moduls*).

Das Branding erscheint in den folgenden Dateien:

icons/weblate.svg

Das Logo wird in der Navigationsleiste angezeigt.

logo-*.png

Web-Symbole je nach Bildschirmauflösung und Webbrowser.

favicon.ico

Von älteren Browsern verwendetes Websymbol.

weblate-*.png

Avatare für Bots oder anonyme Benutzer. Einige Webbrowser verwenden diese als Shortcut-Symbole.

email-logo.png

Wird in Benachrichtigungs-E-Mails verwendet.

2. Zu `INSTALLED_APPS` hinzufügen:

```
INSTALLED_APPS = (  
    # Add your customization as first  
    "weblate_customization",  
    # Weblate apps are here...  
)
```

3. Führen Sie `weblate collectstatic --noinput` aus, um statische Dateien zu sammeln, die an Clients geliefert werden.

Siehe auch:

[How to manage static files](#) (e.g. images, JavaScript, CSS), *Bereitstellung statischer Dateien*

2.22.3 Benutzerdefinierte Qualitätsprüfungen, Erweiterungen und automatische Korrekturen

So installieren Sie Ihren Code für *Benutzerdefinierte automatische Korrekturen*, *Eigene Qualitätsprüfungen schreiben* oder *Schreiben einer Erweiterung* in Weblate:

1. Platzieren Sie die Dateien in Ihr Python-Modul, das die Weblate-Anpassung enthält (siehe *Erstellung eines Python-Moduls*).
2. Fügen Sie den vollqualifizierten Pfad zur Python-Klasse in den entsprechenden Einstellungen (`WEBLATE_ADDONS`, `CHECK_LIST` oder `AUTOFIX_LIST`) hinzu:

```
# Checks  
CHECK_LIST += ("weblate_customization.checks.FooCheck",)  
  
# Autofixes  
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)  
  
# Add-ons  
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Siehe auch:

Benutzerdefinierte automatische Korrekturen, *Eigene Qualitätsprüfungen schreiben*, *Schreiben einer Erweiterung*, *Ausführen von Skripten der Erweiterung*

2.23 Verwaltungsoberfläche

Die Verwaltungsoberfläche bietet Verwaltungseinstellungen unter der URL `/manage/`. Sie ist für Benutzer verfügbar, die mit Administratorrechten angemeldet sind, und kann über das Schraubenschlüssel-Symbol oben rechts aufgerufen werden:

Sie enthält einen grundlegenden Überblick über Ihr Weblate:

- Supportstatus, siehe *Getting support for Weblate*
- Backups, siehe *Sichern und Verschieben von Weblate*
- Gemeinsamer Übersetzungsspeicher, siehe *Speicher*
- Leistungsbericht zur Überprüfung des Zustands von Weblate und der Länge von Celery-Warteschlangen
- SSH-Schlüsselverwaltung, siehe *SSH-Repositorys*
- Übersicht der Alerts für alle Komponenten, siehe *alerts*

2.23.1 Die Django-Adminoberfläche

Warnung: Seien Sie vorsichtig, denn dies ist eine Schnittstelle auf niedriger Ebene. Sie sollten diese in den meisten Fällen nicht benötigen, da die meisten Dinge bequem über die Weblate-Benutzeroberfläche oder die API zugänglich sind.

Hier können Sie in der Datenbank gespeicherte Objekte, wie Benutzer, Übersetzungen und andere Einstellungen, verwalten:

Weblate administration

WELCOME **WEBLATE TEST** RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT

Site administration

REPORTS

Weblate support status

Status of repositories

SSH keys

Performance report

Translation memory

ACCOUNTS

Audit log entries

User profiles

Verified e-mails

AUTH TOKEN

Tokens

AUTHENTICATION

Groups

Roles

Users

BILLING

Billing plans

Customer billings

Invoices

FONTS

Font groups

Fonts

LEGAL

TOS agreements

PERIODIC TASKS

Clocked

Crontabs

Intervals

Periodic tasks

Solar events

PYTHON SOCIAL AUTH

Associations

Nonces

User social auths

SCREENSHOTS

Screenshots

TRANSLATION MEMORY

Translation memory entries

WEBLATE CONFIGURATION

Settings

WEBLATE LANGUAGES

Languages

WEBLATE TRANSLATIONS

Announcements

Component lists

Components

Contributor agreements

Projects

Recent actions

My actions

None available

Im Abschnitt *Auswertungen* können Sie den Status Ihrer Website überprüfen, sie für die *Produktionseinrichtung* optimieren oder SSH-Schlüssel für den *Zugriff auf Repositorys* verwalten.

Verwalten Sie Datenbankobjekte unter einem der Abschnitte. Der interessanteste Bereich ist wahrscheinlich *Weblate-Übersetzungen*, wo Sie übersetzbare Projekte verwalten können, siehe *Projektkonfiguration* und *Komponentenkonfiguration*.

Weblate-Sprachen enthält Sprachdefinitionen, die in *Sprachdefinitionen* näher erläutert werden.

Hinzufügen eines Projekts

Das Hinzufügen eines Projekts dient als Container für alle Komponenten. Normalerweise erstellen Sie ein Projekt für ein Stück Software oder ein Buch (siehe *Projektkonfiguration* für Informationen zu den einzelnen Parametern):

Weblate administration

WELCOME, **WEBLATE TEST** · [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

[Home](#) · [Weblate translations](#) · [Projects](#) · [Add Project](#)

Add Project

Required fields are marked in bold.

Project name:

Display name

URL slug:

Name used in URLs and filenames.

Project website:

Main website of translated project.

Translation instructions:

You can use Markdown and mention users by @username.

☒ Set "Language-Team" header

Lets Weblate update the "Language-Team" file header of your project.

☒ Use shared translation memory

Uses the pool of shared translations between projects.

☒ Contribute to shared translation memory

Contributes to the pool of shared translations between projects.

Access control:

Protected ▾

How to restrict access to this project is detailed in the documentation.

☐ Enable reviews

Requires dedicated reviewers to approve translations.

☐ Enable source reviews

Requires dedicated reviewers to approve source strings.

☒ Enable hooks

Whether to allow updating this repository by remote hooks.

Language aliases:

Comma-separated list of language code mappings, for example: en_GB:en,en_US:en

Machinery settings:

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Siehe auch:

Projektkonfiguration

Zweisprachige Komponenten

Sobald Sie ein Projekt hinzugefügt haben, können Sie ihm Übersetzungskomponenten hinzufügen. (Siehe *Komponentenkonfiguration* für Informationen zu den einzelnen Parametern):

[illegible]

Siehe auch:

Komponentenkonfiguration, Zweisprachige und einsprachige Formate

Einsprachige Komponenten

Um die Übersetzung zu erleichtern, stellen Sie eine Vorlagendatei zur Verfügung, die die Zuordnung der Nachrichten-IDs zur jeweiligen Ausgangssprache (normalerweise Englisch) enthält. (Siehe *Komponentenkonfiguration* für Informationen zu den einzelnen Parametern):

[illegible]

Siehe auch:

Komponentenkonfiguration, Zweisprachige und einsprachige Formate

2.24 Getting support for Weblate

Weblate is copylefted libre software with community support. Subscribers receive priority support at no extra charge. Prepaid help packages are available for everyone. You can find more info about current support offerings at <https://weblate.org/support/>.

2.24.1 Integrating support

Neu in Version 3.8.

Purchased support packages can optionally be integrated into your Weblate [subscription management](#) interface, from where you will find a link to it. Basic instance details about your installation are also reported back to Weblate this way.

The screenshot displays the Weblate web interface. At the top, there's a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with a horizontal menu of tabs: Weblate status (active), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, Teams, Appearance, Tools, Automatic suggestions, and Billing. The 'Weblate status' section is highlighted and contains a table with the following information:

Weblate support status	
Weblate version	4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5
Support status	Community support Refresh support status

Below the table are two buttons: 'Purchase support package' and 'Donate to Weblate'. Underneath is the 'Activate support package' section, which includes a text input field for an 'Activation token' and a note: 'Please enter the activation token obtained when making the subscription.' At the bottom of this section are two buttons: 'Activate' and 'Purchase support package'.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

2.24.2 Data submitted to the Weblate

- URL where your Weblate instance is configured
- Ihr Seitentitel
- The Weblate version you are running
- Tallies of some objects in your Weblate database (projects, components, languages, source strings and users)
- The public SSH key of your instance

Additionally, when *Entdecken Sie Weblate* is turned on:

- List of public projects (name, URL and website)

No other data is submitted.

2.24.3 Integration services

- See if your support package is still valid
- *Von Weblate bereitgestellter Backup-Speicher*
- *Entdecken Sie Weblate*

Hinweis: Purchased support packages are already activated upon purchase, and can be used without integrating them.

2.24.4 Entdecken Sie Weblate

Neu in Version 4.5.2.

Bemerkung: Diese Funktion befindet sich derzeit in einer frühen Betaphase.

Discover Weblate is an opt-in service that makes it easier for users to find Weblate servers and communities. Users can browse registered services on <<https://weblate.org/discover/>>, and find there projects to contribute.

Aufnahme in die Liste

Hinweis: Participating in Discover Weblate makes Weblate submit some information about your server, please see *Data submitted to the Weblate*.

To list your server with an active support subscription (see *Integrating support*) in Discover Weblate all you need to do is turn this on in the management panel:

The screenshot shows the Weblate management interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with a grid of buttons: Weblate status (highlighted), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, Teams, Appearance, Tools, Automatic suggestions, and Billing.

The 'Weblate support status' section displays the following information:

- Weblate version:** 4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5
- Support status:** Community support (with a 'Refresh support status' link)
- Discover Weblate:** Your Weblate is not listed on weblate.org (with an 'Enable discovery' button and a 'Browse discovery' link)

At the bottom of this section are three buttons: 'Manage support package', 'Purchase support package', and 'Donate to Weblate'.

The 'Activate support package' section contains the following text and form:

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

[Text input field]

Please enter the activation token obtained when making the subscription.

At the bottom are two buttons: 'Activate' and 'Purchase support package'.

At the very bottom of the page, a footer line reads: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

Auflistung Ihres Servers ohne ein Support-Abonnement in Discover Weblate:

1. Registrieren Sie sich unter <https://weblate.org/user/>
2. Registrieren Sie Ihren Weblate-Server in der Discovery-Datenbank unter <https://weblate.org/subscription/discovery/>
3. Confirm the service activation in your Weblate and turn on the discovery listing in your Weblate management page using *Enable discovery* button:

W Weblate

DashboardProjects ▾Languages ▾Checks ▾

Manage

Weblate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Teams

Appearance

Tools

Automatic suggestions

Billing

Weblate support status

Weblate version

4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5

Support status

Community support

Refresh support status

Discover Weblate

Your Weblate is not listed on weblate.org

Enable discovery

Browse discovery

Manage support package

Purchase support package

Donate to Weblate

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Auflistung anpassen

Sie können den Eintrag individuell gestalten, indem Sie einen Text und ein Bild (570 × 260 Pixel) unter <https://weblate.org/user/> bereitstellen.

2.25 Rechtliche Dokumente

Bemerkung: Herein you will find various legal information you might need to operate Weblate in certain legal jurisdictions. It is provided as a means of guidance, without any warranty of accuracy or correctness. It is ultimately your responsibility to ensure that your use of Weblate complies with all applicable laws and regulations.

2.25.1 Licensing compliance

Weblate comes with [REUSE 3.0](#) compliant license specification.

2.25.2 ITAR and other export controls

Weblate can be run within your own datacenter or virtual private cloud. As such, it can be used to store ITAR or other export-controlled information, however, end users are responsible for ensuring such compliance.

The Hosted Weblate service has not been audited for compliance with ITAR or other export controls, and does not currently offer the ability to restrict translations access by country.

2.25.3 US encryption controls

Weblate does not contain any cryptographic code, but might be subject export controls as it uses third party components utilizing cryptography for authentication, data-integrity and -confidentiality.

Most likely Weblate would be classified as ECCN 5D002 or 5D992 and, as publicly available libre software, it should not be subject to EAR (see [Encryption items NOT Subject to the EAR](#)).

Software components used by Weblate (listing only components related to cryptographic function):

Python

See https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG

Optionally used by Weblate

Git

Optionally used by Weblate

curl

Used by Git

OpenSSL

Used by Python and cURL

The strength of encryption keys depends on the configuration of Weblate and the third party components it interacts with, but in any decent setup it will include all export restricted cryptographic functions:

- In excess of 56 bits for a symmetric algorithm
- Factorisation of integers in excess of 512 bits for an asymmetric algorithm
- Computation of discrete logarithms in a multiplicative group of a finite field of size greater than 512 bits for an asymmetric algorithm
- Discrete logarithms in a group different than above in excess of 112 bits for an asymmetric algorithm

Weblate doesn't have any cryptographic activation feature, but it can be configured in a way where no cryptography code would be involved. The cryptographic features include:

- Accessing remote servers using secure protocols (HTTPS)
- Generating signatures for code commits (PGP)

Siehe auch:

[Export Controls \(EAR\) on Open Source Software](#)

3.1 Zu Weblate beitragen

Es gibt Dutzende von Möglichkeiten, Weblate zu verbessern. Sie können das auswählen, womit Sie sich wohl fühlen, sei es Programmierung, Grafikdesign, Dokumentation, Sponsoring oder eine Idee:

- *Probleme in Weblate melden*
- *Code für Weblate beisteuern*
- *Mitarbeit an Weblate-Modulen*
- *Weblate übersetzen*
- *Zur Weblate-Dokumentation beitragen*
- *Weblate-Diskussionen*
- *Funding Weblate development*

3.1.1 Weblate übersetzen

Weblate wird laufend mit Hilfe von Weblate selbst *übersetzt* <<https://hosted.weblate.org/>>. Helfen Sie mit, Weblate in so vielen Sprachen wie möglich verfügbar zu machen. Das bringt Weblate näher an seine Benutzer!

Wenn Sie einen möglichen Fehler in der Ausgangszeichenkette finden, können Sie ihn im Weblate-Editor mit einem Kommentar markieren. Auf diese Weise kann er diskutiert und korrigiert werden. Wenn Sie sich sicher sind, können Sie auch auf den Link im Abschnitt *Ort der Ausgangszeichenkette* klicken und einen Pull Request mit Ihrer Korrektur einreichen.

3.1.2 Zur Weblate-Dokumentation beitragen

You are welcome to improve the documentation page of your choice. Do it easily by clicking the *Edit on GitHub* button in the top-right corner of the page.

Please respect these guidelines while writing:

1. Don't remove part of the documentation if it's valid.
2. Use clear and easily-understandable language. You are writing tech docs, not a poem. Not all docs readers are native speakers, be thoughtful.
3. Don't be afraid to ask if you are not certain. If you have to ask about some feature while editing, don't change its docs before you have the answer. This means: You change or ask. Don't do both at the same time.
4. Verify your changes by performing described actions while following the docs.
5. Send PR with changes in small chunks to make it easier and quicker to review and merge.
6. If you want to rewrite and change the structure of a big article, do it in two steps:
 1. Rewrite
 2. Once the rewrite is reviewed, polished, and merged, change the structure of the paragraphs in another PR.

Hinweis: You can [translate the docs](#).

3.1.3 Erweiterung der integrierten Sprachdefinitionen

The language definitions are in the [weblate-language-data repository](#).

Sie können fehlende Sprachdefinitionen gerne in `languages.csv` ergänzen, andere Dateien werden aus dieser Datei generiert.

3.1.4 Weblate-Diskussionen

If you have an idea and not sure if it's suitable for an issue, don't worry. You can join the community in [GitHub discussions](#).

3.1.5 Funding Weblate development

You can boost Weblate's development on the [donate page](#). Funds collected there are used to enable gratis hosting for libre software projects and further development of Weblate. Please check the [donate page](#) for options, such as funding goals and the rewards you get as a proud funder.

Unterstützer, die Weblate finanziert haben

Liste der Weblate-Unterstützer:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- [Cassidy James](#)
- Loic Dachary
- Marozed

- <https://freedombox.org/>
- GNU Solidario (GNU Health)
- BallotReady
- Richard Nespithal
- MyExpenses.Mobi
- Michael Totschnig

Möchten Sie in die Liste aufgenommen werden? Bitte beachten Sie die Optionen auf der Seite [Spenden für Weblate](#).

3.2 Code für Weblate beisteuern

Verstehen Sie den Quellcode von Weblate, indem Sie *Weblate-Quellcode*, *Weblate-Frontend* und *Weblate-Intern* durchgehen.

3.2.1 Mit der Codebasis beginnen

Familiarize yourself with the Weblate codebase, by having a go at the bugs labelled [good first issue](#).

You are welcome to start working on these issues without asking. Just announce that in the issue, so that it's clear that somebody is working on that issue.

3.2.2 Running Weblate locally

The most comfortable approach to get started with Weblate development is to follow *Installieren aus Quellen*. It will get you a virtualenv with editable Weblate sources.

1. Klonen des Weblate-Quellcodes:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Erstellen Sie eine virtualenv:

```
virtualenv .venv
.venv/bin/activate
```

3. Install Weblate (for this you need some system dependencies, see *Installieren aus Quellen*):

```
pip install -e .
```

3. Install all dependencies useful for development:

```
pip install -r requirements-dev.txt
```

4. Start a development server:

```
weblate runserver
```

5. Depending on your configuration, you might also want to start Celery workers:

```
./weblate/examples/celery start
```

6. To run a test (see *Local testing* for more details):

```
. scripts/test-database.sh
./manage.py test
```

Siehe auch:

Installieren aus Quellen

3.2.3 Running Weblate locally in Docker

If you have Docker and docker-compose installed, you can spin up the development environment by simply running:

```
./rundev.sh
```

It will create a development Docker image and start it. Weblate is running on <http://127.0.0.1:8080/> and you can sign in as the user `admin` using `admin` as the password. The new installation is empty, so you might want to continue with *Adding translation projects and components*.

The Dockerfile and docker-compose.yml for this are located in the `dev-docker` directory.

Das Skript akzeptiert auch einige Parameter, um Tests auszuführen, führen Sie es mit dem Parameter `test` aus und geben Sie dann alle `test`-Parameter an, um zum Beispiel nur Tests im Modul `weblate.machine` auszuführen:

```
./rundev.sh test --failfast weblate.machine
```

Bemerkung: Be careful that your Docker containers are up and running before running the tests. You can check that by running the `docker ps` command.

To display the logs:

```
./rundev.sh logs
```

To stop the background containers, run:

```
./rundev.sh stop
```

Wenn Sie das Skript ohne Argumente ausführen, wird der Docker-Container neu erstellt und neu gestartet.

Bemerkung: This is not a suitable setup for production, as it includes several hacks which are insecure, but they make development easier.

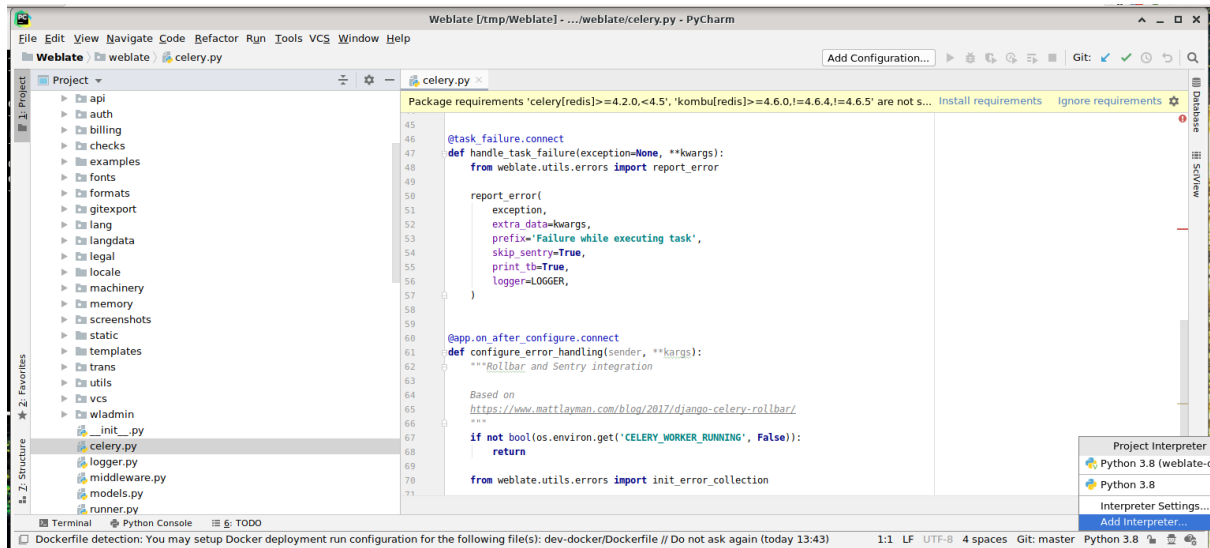
3.2.4 Bootstrapping your devel instance

You might want to use `import_demo` to create demo translations and `createadmin` to make an admin user.

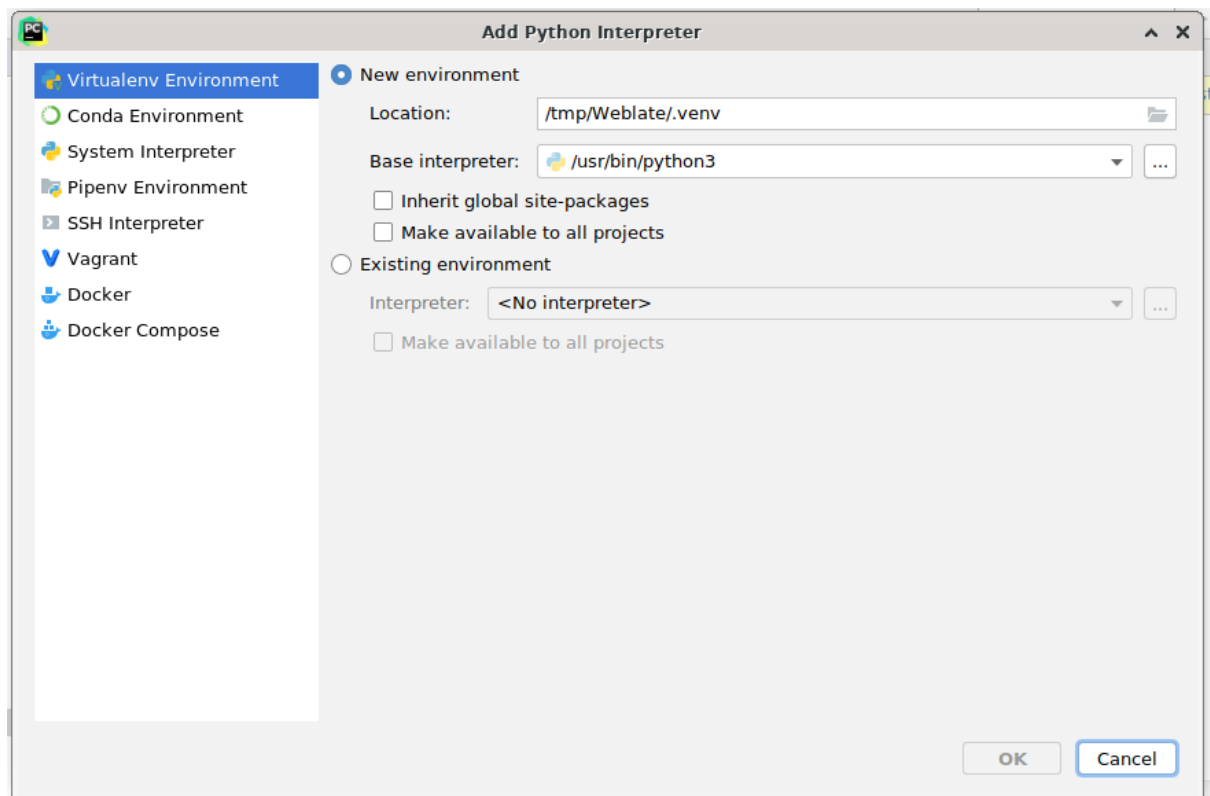
3.2.5 Coding Weblate with PyCharm

PyCharm is a known IDE for Python, here are some guidelines to help you set up your Weblate project in it.

Considering you have just cloned the GitHub repository to a folder, just open it with PyCharm. Once the IDE is open, the first step is to specify the interpreter you want to use:

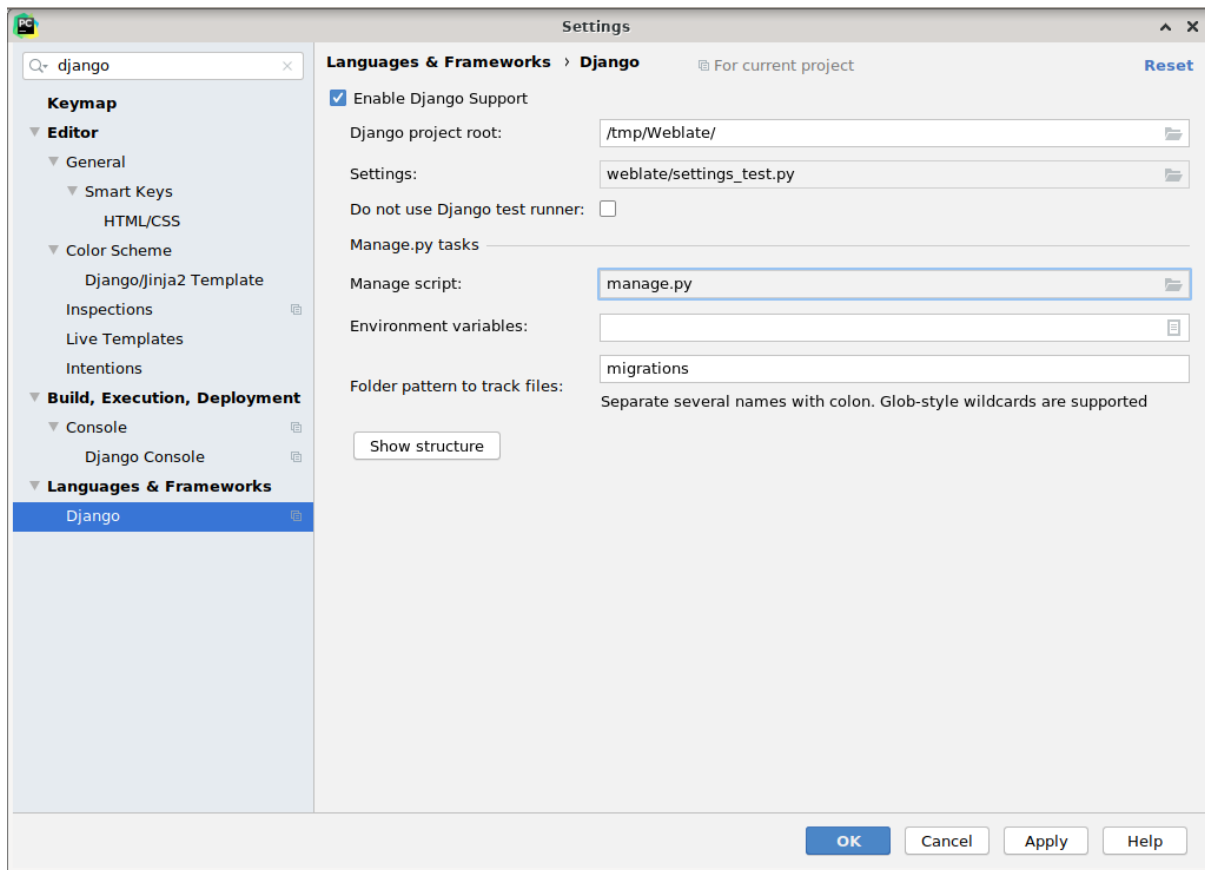


You can either choose to let PyCharm create the virtualenv for you, or select an already existing one:



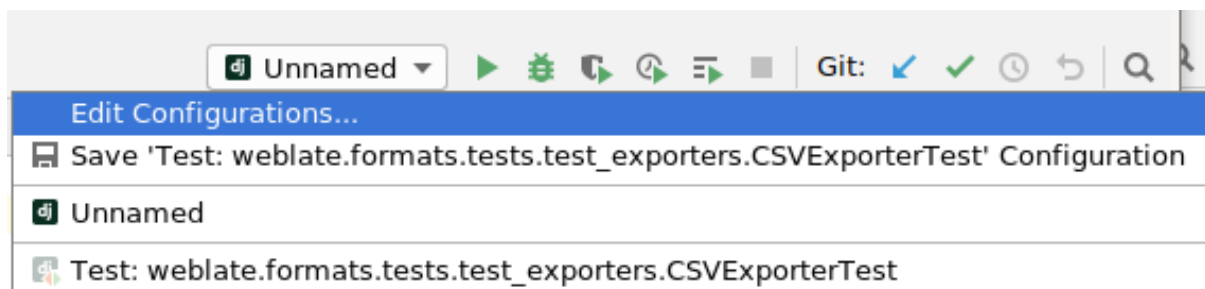
Vergessen Sie nicht, die Abhängigkeiten zu installieren, sobald der Interpreter eingestellt ist: Entweder über die Konsole (die Konsole der IDE verwendet standardmäßig direkt Ihre virtualenv), oder über die Benutzeroberfläche, wenn Sie eine Warnung über fehlende Abhängigkeiten erhalten.

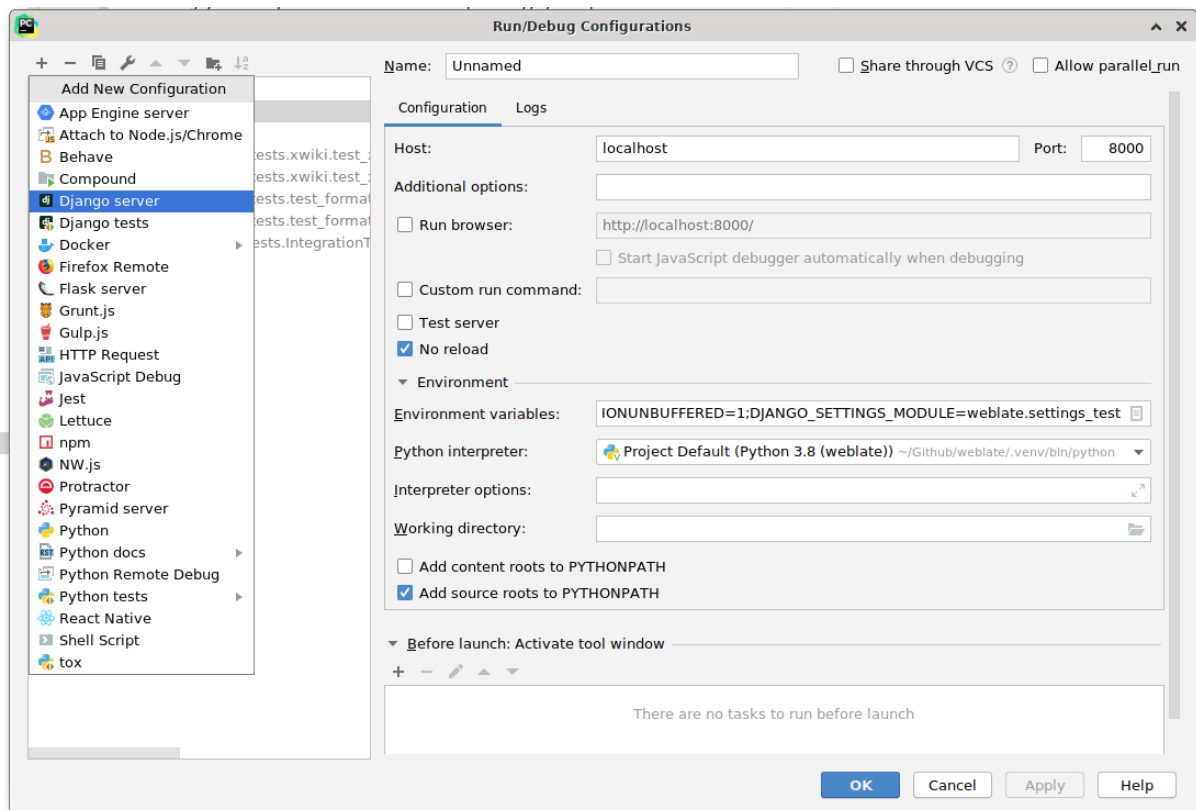
The second step is to set the right info to use Django natively inside PyCharm: The idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of the Django project and the path to its settings:



Be careful, the *Django project root* is the actual root of the repository, not the Weblate sub-directory. About the settings, you could use the `weblate/settings_test.py` from the repository, but you could create your own setting and set it there.

The last step is to run the server and to put breakpoints in the code to be able to debug it. This is done by creating a new *Django Server* configuration:





Hinweis: Seien Sie vorsichtig mit der Eigenschaft namens *No reload*: Sie verhindert, dass der Server live neu geladen wird, wenn Sie Dateien ändern. Dadurch bleiben die vorhandenen Debugger-Haltepunkte erhalten, obwohl sie normalerweise beim Neuladen des Servers verworfen würden.

3.3 Weblate-Quellcode

Weblate wird auf [GitHub](#) entwickelt. Sie sind herzlich eingeladen, den Code zu forken und Pull Requests zu erstellen. Patches in jeder anderen Form sind ebenfalls willkommen.

Siehe auch:

Check out [Weblate-Interna](#) to see how Weblate looks from inside.

3.3.1 Coding guidelines

Any code for Weblate should be written with [Security by Design Principles](#) in mind.

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize [PEP 484](#) type hints. We're not checking this in our CI yet as existing code does not yet include them.

3.3.2 Coding standard and linting the code

The code should follow PEP-8 coding guidelines and should be formatted using **black** code formatter.

To check the code quality, you can use **flake8**, the recommended plugins are listed in `.pre-commit-config.yaml` and its configuration is placed in `setup.cfg`.

The easiest approach to enforce all this is to install **pre-commit**. The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the `requirements-lint.txt`) turn it on by running `pre-commit install` in Weblate checkout. This way all your changes will be automatically checked.

You can also trigger check manually, to check all files run:

```
pre-commit run --all
```

3.4 Fehlersuche in Weblate

Fehler können sich in Form von Anwendungsabstürzen oder verschiedenen Fehlverhaltensweisen äußern. Sie sind herzlich eingeladen, Informationen zu solchen Problemen zu sammeln und sie an den [Issue Tracker](#) zu übermitteln.

3.4.1 Debugmodus

Wenn Sie den Debugmodus aktivieren, werden die Ausnahmen im Webbrowser angezeigt. Dies ist nützlich, um Probleme in der Weboberfläche zu beheben, eignet sich aber nicht für eine Produktionsumgebung, da es Auswirkungen auf die Leistung hat und möglicherweise private Daten preisgibt.

In a production environment, use [ADMINS](#) to receive e-mails containing error reports, or configure error collection using a third-party service.

Siehe auch:

[Debugmodus deaktivieren](#), [Administratoren richtig konfigurieren](#), [Sammeln von Fehlerberichten](#)

3.4.2 Weblate logs

Weblate can produce detailed logs of what is going on in the background. In the default configuration it uses syslog and that makes the log appear either in `/var/log/messages` or `/var/log/syslog` (depending on your syslog daemon configuration).

The Celery process (see [Hintergrundaufgaben mit Celery](#)) usually produces its own logs as well. The example system-wide setups logs to several files under `/var/log/celery/`.

Docker containers log to their output (as per usual in the Docker world), so you can look at the logs using `docker-compose logs`.

Siehe auch:

[Sample configuration](#) enthält die `LOGGING` Konfiguration.

3.4.3 Not processing background tasks

A lot of things are done in the background by Celery workers. If things like sending out e-mails or component removal does not work, there might be a related issue.

Dinge, die in diesem Fall zu überprüfen sind:

- Überprüfen Sie, ob der Celery-Prozess ausgeführt wird, siehe *Hintergrundaufgaben mit Celery*
- Check the Celery queue status, either in *Verwaltungsoberfläche*, or using `celery_queues`
- Look in the Celery logs for errors (see *Weblate logs*)

3.4.4 Keine E-Mails von Weblate erhalten

You can verify whether outgoing e-mail is working correctly by using the `sendtestemail` management command (see *Invoking management commands* for instructions on how to invoke it in different environments) or by using *Verwaltungsoberfläche* under the *Tools* tab.

These send e-mails directly, so this verifies that your SMTP configuration is correct (see *Konfigurieren ausgehender E-Mail*). Most of the e-mails from Weblate are however sent in the background and there might be some issues with Celery involved as well, please see *Not processing background tasks* for debugging that.

3.4.5 Analyse von Anwendungsabstürzen

In case the application crashes, it is useful to collect as much info about the crash as possible. This can be achieved by using third-party services which can collect such info automatically. You can find info on how to set this up in *Sammeln von Fehlerberichten*.

3.4.6 Stille Ausfälle

Lots of tasks are offloaded to Celery for background processing. Failures are not shown in the user interface, but appear in the Celery logs. Configuring *Sammeln von Fehlerberichten* helps you to notice such failures easier.

3.4.7 Leistungsprobleme

In case Weblate performs badly in some scenario, please collect the relevant logs showing the issue, and anything that might help figuring out where the code might be improved.

In case some requests take too long without any indication, you might want to install `dogslow` along with *Sammeln von Fehlerberichten* and get pinpointed and detailed tracebacks in the error collection tool.

Falls die langsame Leistung mit der Datenbank zusammenhängt, können Sie auch die Protokollierung aller Datenbankabfragen aktivieren, indem Sie nach der Aktivierung von *DEBUG* die folgende Konfiguration verwenden:

```
LOGGING["loggers"]["django.db.backends"] = {"handlers": ["console"], "level":  
↪ "DEBUG" }
```


3.5 Weblate-Interna

Bemerkung: Dieses Kapitel gibt Ihnen einen grundlegenden Überblick über die Interna von Weblate.

Weblate leitet den größten Teil seiner Codestruktur von [Django](#) ab und basiert auf diesem.

3.5.1 Struktur des Verzeichnisses

Schneller Überblick über die Verzeichnisstruktur des Weblate-Hauptrepositorys:

docs

Quellcode für diese Dokumentation, die mit [Sphinx](#) erstellt werden kann.

dev-docker

Docker-Code zum Betrieb des Entwicklungsservers, siehe [Running Weblate locally in Docker](#).

weblate

Quellcode von Weblate als [Django](#)-Anwendung, siehe [Weblate-Interna](#).

weblate/static

Client-Dateien (CSS, Javascript und Bilder), siehe [Weblate-Frontend](#).

3.5.2 Module

Weblate besteht aus mehreren Django-Anwendungen (einige sind optional, siehe [Optionale Weblate-Module](#)):

accounts

Benutzerkonto, Profile und Benachrichtigungen.

addons

Erweiterungen um das Verhalten von Weblate zu verändern, siehe [Erweiterungen](#).

api

API basierend auf [Django REST framework](#).

auth

Authentifizierung und Berechtigungen.

billing

Das optionale Modul [Abrechnung](#).

checks

Translation string [Qualitätsprüfungen](#) module.

fonts

Font rendering checks module.

formats

File format abstraction layer based on translate-toolkit.

gitexport

Das optionale Modul [Git-Exporter](#).

lang

Modul zur Definition von Sprach- und Pluralmodellen.

legal

Das optionale Modul *Rechtliche Grundlagen*.

machinery

Integration von maschinellen Übersetzungsdiensten.

memory

Integrierter Übersetzungsspeicher, siehe *Übersetzungsspeicher*.

screenshots

Verwaltung von Bildschirmfotos und OCR-Modul.

trans

Hauptmodul für Übersetzungen.

utils

Verschiedene Hilfsprogramme.

vcs

Abstraktion der Versionsverwaltung.

wladmin

Anpassung der Django-Adminoberfläche.

3.6 Erweiterungen entwickeln

Erweiterungen sind eine Möglichkeit, den Lokalisierungsprozess in Weblate anzupassen.

class `weblate.addons.base.BaseAddon` (*storage=None*)

Base class for Weblate add-ons.

classmethod `can_install` (*component, user*)

Überprüfen Sie, ob die Erweiterung mit der angegebenen Komponente kompatibel ist.

configure (*settings*)

Konfiguration speichern.

daily (*component*)

Hook wird täglich ausgelöst.

classmethod `get_add_form` (*user, component, **kwargs*)

Konfigurationsformular zum Hinzufügen einer neuen Erweiterung zurückgeben.

get_settings_form (*user, **kwargs*)

Konfigurationsformular für diese Erweiterung zurückgeben.

post_add (*translation*)

Hook wird ausgelöst, nachdem eine neue Übersetzung hinzugefügt wurde.

post_commit (*component*)

Hook wird ausgelöst, nachdem Änderungen an das Repository committet wurden.

post_push (*component*)

Hook wird ausgelöst, nachdem das Repository Upstream gepusht wurde.

post_update (*component*, *previous_head*: *str*, *skip_push*: *bool*)

Hook wird ausgelöst, nachdem das Repository von Upstream aktualisiert wurde.

Parameter

- **previous_head** (*str*) – HEAD of the repository prior to update, can be blank on initial clone.
- **skip_push** (*bool*) – Ob die Erweiterungsoperation das Pushen von Änderungen im Upstream überspringen soll. Normalerweise können Sie dies den zugrunde liegenden Methoden als `commit_and_push` oder `commit_pending` übergeben.

pre_commit (*translation*, *author*)

Hook wird ausgelöst, bevor Änderungen an das Repository committet werden.

pre_push (*component*)

Hook wird ausgelöst, bevor das Repository Upstream gepusht wird.

pre_update (*component*)

Hook wird ausgelöst, bevor das Repository von Upstream aktualisiert wird.

save_state ()

Speichern Sie Informationen über den Status der Erweiterung.

store_post_load (*translation*, *store*)

Hook wird ausgelöst, nachdem eine Datei analysiert wurde.

It receives an instance of a file format class as a argument.

This is useful to modify file format class parameters, for example adjust how the file will be saved.

unit_pre_create (*unit*)

Hook wird ausgelöst, bevor eine neue Einheit erstellt wird.

Hier ein Beispiel für eine Erweiterung:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT

class ExampleAddon(BaseAddon):
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {"file_format": {"po", "po-mono"}}
    # List of events add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
    # Detailed add-on description
    description = _("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Weblate-Frontend

Das Frontend wird derzeit mit Bootstrap, jQuery und einigen Bibliotheken von Drittanbietern erstellt.

3.7.1 Unterstützte Browser

Weblate unterstützt die neuesten, stabilen Versionen aller gängigen Browser und Plattformen.

Alternative Browser, welche die neueste Version von WebKit, Blink oder Gecko verwenden, ob direkt oder über die Web View API der Plattform, werden nicht explizit unterstützt. Weblate sollte jedoch (in den meisten Fällen) auch in diesen Browsern korrekt angezeigt werden und funktionieren.

Ältere Browser könnten funktionieren, aber einige Funktionen sind möglicherweise eingeschränkt.

3.7.2 Verwaltung von Abhängigkeiten

Der Yarn-Paketmanager wird verwendet, um Bibliotheken von Drittanbietern zu aktualisieren. Die Konfiguration befindet sich in `scripts/yarn` und es gibt ein Wrapper-Skript `scripts/yarn-update`, um die Bibliotheken zu aktualisieren, sie zu bauen und an die richtigen Stellen in `weblate/static/vendor` zu kopieren, wo sich der gesamte Frontend-Code von Drittanbietern befindet. Der Weblate-spezifische Code sollte direkt in `weblate/static` oder in funktionspezifische Unterverzeichnisse (zum Beispiel `weblate/static/editor`) kopiert werden.

Das Hinzufügen einer neuen Bibliothek eines Drittanbieters besteht normalerweise aus:

```
# Add a yarn package
yarn --cwd scripts/yarn add PACKAGE
# Edit the script to copy package to the static folder
edit scripts/yarn-update
# Run the update script
./scripts/yarn-update
# Add files to git
git add .
```

3.7.3 Programmierstil

Weblate verwendet [Prettier](#) für die Code-Formatierung von JavaScript- und CSS-Dateien.

Außerdem verwenden wir [ESLint](#), um den JavaScript-Code zu überprüfen.

3.7.4 Lokalisierung

Sollten Sie einen für den Benutzer sichtbaren Text im Frontend-Code benötigen, sollte dieser lokalisierbar sein. In den meisten Fällen genügt es, den Text in die Funktion `gettext` einzuschließen, aber es sind auch komplexere Funktionen verfügbar:

```
document.write(gettext('this is to be translated'));

var object_count = 1 // or 0, or 2, or 3, ...
s = ngettext('literal for the singular case',
            'literal for the plural case', object_count);

fmts = ngettext('There is %s object. Remaining: %s',
                'There are %s objects. Remaining: %s', 11);
s = interpolate(fmts, [11, 20]);
// s is 'There are 11 objects. Remaining: 20'
```

Siehe auch:

Übersetzungsthema in der Django-Dokumentation

3.7.5 Symbole

Weblate verwendet derzeit Material Design Symbole. Falls Sie auf der Suche nach neuen Symbolen sind, schauen Sie unter [Material Design Icons](#) oder [Material Design Resources](#).

Zusätzlich gibt es `scripts/optimize-svg`, um die Größe der SVG zu reduzieren, da die meisten Symbole in das HTML eingebettet sind, um die Gestaltung der Pfade zu ermöglichen.

3.8 Probleme in Weblate melden

Weblate [Issue Tracker](#) wird auf GitHub gehostet.

Sie können dort gerne alle Probleme melden, die Sie haben, oder Verbesserungsvorschläge für Weblate machen. Es gibt verschiedene Vorlagen, die Sie bequem durch den Problembericht führen.

Wenn Sie ein Sicherheitsproblem in Weblate gefunden haben, lesen Sie bitte den Abschnitt Sicherheit unten.

Wenn Sie sich nicht sicher sind, ob es sich um einen Fehlerbericht oder Vorschlag für eine neue Funktion handelt, können Sie Diskussionen versuchen.

3.8.1 Sicherheitslücken

Um der Community Zeit zu geben, zu reagieren und zu aktualisieren, werden Sie dringend gebeten, alle Sicherheitsprobleme privat zu melden. HackerOne wird verwendet, um Sicherheitsprobleme zu behandeln, und kann direkt unter [HackerOne](#) gemeldet werden. Sobald Sie das Problem dort gemeldet haben, hat die Gemeinschaft nur eine begrenzte, aber ausreichende Zeit, um den Vorfall zu lösen.

Alternativ können Sie sich auch an security@weblate.org wenden, das ebenfalls auf HackerOne veröffentlicht wird.

Wenn Sie HackerOne, aus welchen Gründen auch immer, nicht verwenden wollen, können Sie den Bericht per E-Mail an michal@weblate.org senden. Sie können ihn mit diesem PGP-Schlüssel `3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D` verschlüsseln. Sie können den PGP-Schlüssel auch von [Keybase](#) beziehen.

Bemerkung: Weblate ist für viele Dinge von Komponenten Dritter abhängig. Falls Sie eine Schwachstelle finden, die eine dieser Komponenten betrifft, melden Sie diese bitte direkt an das entsprechende Projekt.

Einige davon sind:

- [Django](#)
 - [Django REST-Framework](#)
 - [Python Social Auth](#)
-

3.9 Weblate-Testsuite und kontinuierliche Integration

Testsuites exist for most of the current code, increase coverage by adding testcases for any new functionality, and verify that it works.

3.9.1 Continuous integration

Current test results can be found on [GitHub Actions](#) and coverage is reported on [Codecov](#).

There are several jobs to verify different aspects:

- Unit tests
- Documentation build and external links
- Migration testing from all supported releases
- Code linting
- Setup verification (ensures that generated dist files do not miss anything and can be tested)

The configuration for the CI is in `.github/workflows` directory. It heavily uses helper scripts stored in `ci` directory. The scripts can be also executed manually, but they require several environment variables, mostly defining Django settings file to use and database connection. The example definition of that is in `scripts/test-database.sh`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

# Simple way to configure test database from environment

# shellcheck shell=sh

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE="${1:-postgresql}"

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

The simple execution can look like:

```
. scripts/test-database.sh
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Local testing

To run a testsuite locally, use:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Hinweis: You will need a database (PostgreSQL) server to be used for tests. By default Django creates separate database to run tests with `test_` prefix, so in case your settings is configured to use `weblate`, the tests will use `test_weblate` database. See *Datenbankeinrichtung für Weblate* for setup instructions.

The `weblate/settings_test.py` is used in CI environment as well (see *Continuous integration*) and can be tuned using environment variables:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

# Simple way to configure test database from environment

# shellcheck shell=sh

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE="${1:-postgresql}"

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

Prior to running tests you should collect static files as some tests rely on them being present:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

You can also specify individual tests to run:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Hinweis: The tests can also be executed inside developer docker container, see *Running Weblate locally in Docker*.

Siehe auch:

Siehe *Testing in Django* für weitere Informationen zum Ausführen und Schreiben von Tests für Django.

3.10 Datenschemata

Weblate verwendet *JSON-Schema*, um das Layout von externen JSON-Dateien zu definieren.

3.10.1 Weblate Übersetzungsspeicher-Schema

https://weblate.org/schemas/weblate-memory.schema.json		
type	array	
items	Das Übersetzungsspeicher-Element	
	type	object
	properties	
	• category	Die Zeichenkette-Kategorie
		1 ist global, 2 ist gemeinsam, 10000000+ sind projekt-spezifisch, 20000000+ sind benutzerspezifisch
		type integer
		examples 1
		minimum 0
		default 1
	• origin	Der Zeichenkette-Ursprung
		Dateiname oder Komponentennamenname
		type String
		examples test.tmx
		project/component
		default
	• source	Die Ausgangszeichenkette
		type String
		examples Hello
		minLength 1
		default
	• source_language	Die Ausgangssprache
		ISO 639-1 / ISO 639-2 / IETF BCP 47
		type String
		examples en
		pattern ^[^\s]+\$
		default
	• target	Die Zielzeichenkette
		type String
		examples Ahoj
		minLength 1
		default
	• target_language	Die Zielsprache
		ISO 639-1 / ISO 639-2 / IETF BCP 47
		type String
		examples cs
		pattern ^[^\s]+\$
		default
	additionalProperties	False
Definitionen		

Siehe auch:

Übersetzungsspeicher, *dump_memory*, *import_memory*

3.10.2 Weblate Benutzerdatenexport

https://weblate.org/schemas/weblate-userdata.schema.json		
type	<i>object</i>	
properties		
• basic	<i>Basis</i>	
	type	<i>object</i>
	properties	
	• username	<i>Benutzername</i>
		type <i>String</i>
		examples <code>admin</code>
		default
	• full_name	<i>Vollständiger Name</i>
		type <i>String</i>
		examples <code>Weblate-Administrator</code>
		default
	• email	<i>E-Mail</i>
		type <i>String</i>
		examples <code>noreply@example.com</code>
		default
		format <code>email</code>
	• date_joined	<i>Datum des Beitritts</i>
		type <i>String</i>
		examples <code>2019-11-18T18:53:54.862Z</code>
		default
		format <code>date-time</code>
	additionalProperties	<code>False</code>
• profile	<i>Profil</i>	
	type	<i>object</i>
	properties	
	• language	<i>Sprache</i>
		type <i>String</i>
		examples <code>cs</code>
		pattern <code>^[^]*\$</code>
		default
	• suggested	<i>Anzahl vorgeschlagener Zeichenketten</i>
		type <i>integer</i>
		examples <code>1</code>
		default <code>0</code>
	• translated	<i>Anzahl übersetzter Zeichenketten</i>
		type <i>integer</i>
		examples <code>24</code>
		default <code>0</code>
	• uploaded	<i>Anzahl hochgeladener Bildschirmfotos</i>
		type <i>integer</i>
		examples <code>1</code>
		default <code>0</code>
	• hide_completed	<i>Abgeschlossene Übersetzungen in der Übersicht ausblenden</i>
		type <i>boolean</i>
		examples <code>False</code>
		default <code>True</code>
	• secondary_in_zen	<i>Zweitsprachen im Zen-Modus anzeigen</i>
		type <i>boolean</i>
		examples <code>True</code>
		default <code>True</code>

Fortsetzung auf der nächsten Seite

Tab. 2 – Fortsetzung der vorherigen Seite

• hide_source_sec	<i>Ausgangssprache ausblenden, wenn bereits eine Übersetzung vorhanden ist</i>		
	type	<i>boolean</i>	
	examples	False	
	default	True	
• editor_link	<i>Editor-URL</i>		
	type	<i>String</i>	
	examples		
	pattern	<i>^.*\$</i>	
• translate_mode	<i>Übersetzungseditor-Modus</i>		
	type	<i>integer</i>	
	examples	0	
	default	0	
• zen_mode	<i>Zen-Editor-Modus</i>		
	type	<i>integer</i>	
	examples	0	
	default	0	
• special_chars	<i>Sonderzeichen</i>		
	type	<i>String</i>	
	examples		
	pattern	<i>^.*\$</i>	
• dashboard_view	<i>Standardansicht der Übersichtsseite</i>		
	type	<i>integer</i>	
	examples	1	
	default	0	
• dashboard_components	<i>Voreingestellte Auflistung der Komponenten</i>		
	default	null	
	anyOf	type	<i>null</i>
		type	<i>integer</i>
• languages	<i>Übersetzte Sprachen</i>		
	type	<i>array</i>	
	default		
	items	<i>Sprachcode</i>	
		type	<i>String</i>
		examples	cs
		pattern	<i>^.*\$</i>
• secondary_languages	<i>Zweitsprachen</i>		
	type	<i>array</i>	
	default		
	items	<i>Sprachcode</i>	
		type	<i>String</i>
		examples	sk
		pattern	<i>^.*\$</i>
• watched	<i>Beobachtete Projekte</i>		
	type	<i>array</i>	
	default		
	items	<i>Projekt-Slug</i>	
		type	<i>String</i>
		examples	weblate
		pattern	<i>^.*\$</i>
	default		

Fortsetzung auf der nächsten Seite

Tab. 2 – Fortsetzung der vorherigen Seite

• auditlog	additionalProperties	False	
	<i>Audit-Protokoll</i>		
	type	array	
	default		
	items	<i>Elemente</i>	
	type	object	
	properties		
	• address	<i>IP-Adresse</i>	
	type	String	
	examples	127.0.0.1	
	pattern	^.*\$	
	default		
	• user_agent	<i>User Agent</i>	
	type	String	
	examples	PC / Linux / Firefox 70.0	
	pattern	^.*\$	
	default		
	• timestamp	<i>Zeitstempel</i>	
	type	String	
	examples	2019-11- 18T18:58:30.845Z	
	default		
	format	date-time	
	• activity	<i>Aktivität</i>	
	type	String	
	examples	login	
	pattern	^.*\$	
	default		
	additionalProperties	False	
Definitionen			

Siehe auch:*Benutzerprofil, dumpuserdata*

3.11 Weblate-Veröffentlichungen

3.11.1 Veröffentlichungszeitplan

Weblate hat einen zweimonatigen Veröffentlichungszyklus für neue Versionen (x.y). Darauf folgen normalerweise eine Reihe von Veröffentlichungen zur Behebung von Fehlern, die sich eingeschlichen haben (x.y.z).

Die Änderung der Hauptversion zeigt an, dass der Upgrade-Prozess diese Version nicht überspringen kann – Sie müssen immer auf x.0 aktualisieren, bevor Sie auf höhere x.y-Versionen aktualisieren.

Siehe auch:*Upgrade von Weblate*

3.11.2 Veröffentlichungsplanung

Die Funktionen für kommende Versionen werden mit Hilfe von GitHub-Meilensteinen gesammelt, Sie können unsere Roadmap unter <https://github.com/WeblateOrg/weblate/milestones> einsehen.

3.11.3 Veröffentlichungsprozess

Vor Veröffentlichung zu prüfende Dinge:

1. Neu übersetzte Sprachen überprüfen `./scripts/list-translated-languages`.
2. Endgültige Version einstellen `./scripts/prepare-release`.
3. Bildschirmfotos auf den neuesten Stand bringen `make -j 12 -C docs update-screenshots`.
4. Merge aller möglicherweise ausstehenden Übersetzungen `wlc push; git remote update; git merge origin/weblate`

Die Veröffentlichung ausführen:

5. Ein Release erstellen `./scripts/create-release --tag` (siehe unten für die Anforderungen).

Manuelle Schritte nach der Veröffentlichung:

6. Das Docker-Image aktualisieren.
7. Den GitHub-Meilenstein schließen.
8. Sobald das Docker-Image getestet ist, fügen Sie ein Tag hinzu und pushen es.
9. Das Helm-Diagramm auf die neue Version aktualisieren.
10. Die neue Version in `.github/workflows/migrations.yml` einfügen, um sie bei Migrationstests zu berücksichtigen.
11. Die Version in den Download-Links auf der Website erhöhen.
12. Die Version im Repository mit `./scripts/set-version` erhöhen.
13. Überprüfen, ob readthedocs.org alle Übersetzungen der Dokumentation mit `./scripts/rtd-projects` erstellt hat.

Um Tags mit dem Skript `./scripts/create-release` zu erstellen, wird folgendes benötigt:

- GnuPG mit privatem Schlüssel, der zum Signieren der Veröffentlichung verwendet wird
- Push-Zugriff auf die Git-Repositorys von Weblate (es werden Tags gepusht)
- Konfiguriertes **hub**-Tool und Zugriff um Veröffentlichungen auf dem Weblate-Repository zu erstellen
- SSH-Zugang zum Weblate-Download-Server (die Website-Downloads werden dorthin kopiert)

3.12 Sicherheit und Datenschutz

Tipp: Weblate legt großen Wert auf die Sicherheit und den Schutz der Privatsphäre seiner Nutzer.

Die Entwicklung von Weblate richtet sich nach den [Best Practices der Core-Infrastruktur-Initiative der Linux Foundation](#).

Siehe auch:

[Sicherheitslücken](#)

3.12.1 Sicherheitsupdates

Only the latest release is guaranteed to receive security updates.

3.12.2 Nachverfolgung der Abhängigkeiten auf Schwachstellen

Sicherheitslücken in unseren Abhängigkeiten werden mit [Dependabot](#) überwacht. Dies deckt die Python- und JavaScript-Bibliotheken ab, und die Abhängigkeiten der neuesten stabilen Version wurden aktualisiert, um Sicherheitslücken zu vermeiden.

Hinweis: Es kann Schwachstellen in Bibliotheken von Drittanbietern geben, die Weblate nicht betreffen, so dass diese nicht durch Veröffentlichungen zur Fehlerbehebung von Weblate behoben werden.

3.12.3 Sicherheit von Docker-Containern

Die Docker-Container werden regelmäßig mit den Sicherheitsscannern [Anchore](#) und [Trivy](#) gescannt.

So können wir Schwachstellen frühzeitig erkennen und Verbesserungen schnell veröffentlichen.

Sie können die Ergebnisse dieser Scans auf GitHub abrufen – sie werden als Artefakte auf unserer CI im SARIF-Format (Static Analysis Results Interchange Format) gespeichert.

Siehe auch:

[Continuous integration](#)

3.13 Mitarbeit an Weblate-Modulen

Besides the main repository, Weblate consists of several Python modules. All these follow same structure and this documentation covers them all.

For example, this covers:

- [wlc](#), Python client library, see *[Weblate-Client](#)*
- [translation-finder](#), used to discover translatable files in the repository
- [language-data](#), language definitions for Weblate, see *[Sprachdefinitionen](#)*

3.13.1 Coding guidelines

Any code for Weblate should be written with [Security by Design Principles](#) in mind.

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize [PEP 484](#) type hints. We're not checking this in our CI yet as existing code does not yet include them.

3.13.2 Running tests

The tests are executed using `py.test`. First you need to install test requirements:

```
pip install -r requirements-test.txt
```

You can then execute the testsuite in the repository checkout:

```
py.test
```

Siehe auch:

The CI integration is very similar to *Weblate-Testsuite und kontinuierliche Integration*.

3.13.3 Coding standard and linting the code

The code should follow PEP-8 coding guidelines and should be formatted using **black** code formatter.

To check the code quality, you can use **flake8**, the recommended plugins are listed in `.pre-commit-config.yaml` and its configuration is placed in `setup.cfg`.

The easiest approach to enforce all this is to install `pre-commit`. The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the `requirements-lint.txt`) turn it on by running `pre-commit install` in Weblate checkout. This way all your changes will be automatically checked.

You can also trigger check manually, to check all files run:

```
pre-commit run --all
```

Siehe auch:

Weblate-Quellcode

3.14 Über Weblate

3.14.1 Projektziele

Web-based continuous localization tool with tight *Integration der Versionsverwaltung* supporting a wide range of *file formats*, making it easy for translators to contribute.

3.14.2 Projektname

„Weblate“ ist ein Kofferwort aus den Wörtern „web“ und „translate“.

3.14.3 Projektseite

Die Einstiegsseite ist <https://weblate.org> und es gibt einen Cloud-gehosteten Dienst unter <https://hosted.weblate.org>. Die Dokumentation kann unter <https://docs.weblate.org> nachgelesen werden.

3.14.4 Projektlogos

Die Projektlogos und andere Grafiken sind unter <https://github.com/WeblateOrg/graphics> verfügbar.

3.14.5 Leitung

Dieses Projekt wird von Michal Čihař betreut, den Sie unter michal@weblate.org erreichen können.

3.14.6 Autoren

Weblate wurde von Michal Čihař gegründet. Seit seiner Gründung im Jahr 2012 haben Tausende von Menschen dazu beigetragen.

3.15 Lizenz

Ausführlichere Informationen zur Lizenzierung sind im Weblate-Quellcode verfügbar und folgen der [REUSE 3.0-Spezifikation](#).

Copyright © Michal Čihař michal@weblate.org

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see [<https://www.gnu.org/licenses/>](https://www.gnu.org/licenses/).

4.1 Weblate 4.16

Veröffentlicht am 1. März 2023.

- Format string checks now also detects duplicated formats.
- Die Suchleistung für einige speziell formatierte Zeichenketten verbessert.
- Celery beat is now storing the tasks schedule in the database.
- Unterstützung für IBM Watson Language Translator hinzugefügt.
- Dropped support for VCS integration settings deprecated in 4.14.
- Unterstützung für Bitbucket-Server-Pull-Requests hinzugefügt.
- Verbesserte Konfliktbehandlung in gettext PO-Dateien.
- Added support for defining strings state when adding via API.
- Added support for configuring CORS allowed origins.
- Added plurals support to automatic suggestions.

[Alle Änderungen im Detail.](#)

4.2 Weblate 4.15.2

Veröffentlicht am 25. Januar 2023.

- Enabled gettext JSON and i18next v4 formats in the default configuration.
- Fixed crash on uploading corrupted files.
- Show stale directories in Git repository status.

[Alle Änderungen im Detail.](#)

4.3 Weblate 4.15.1

Veröffentlicht am 19. Januar 2023.

- Die Übersetzungsvorschläge der automatischen Übersetzung korrigiert.
- Absturz der Erweiterungsseite in einigen Ausnahmefällen behoben.
- In einigen Fällen wurde die Vorlage für neue Übersetzungen nicht mehr übersetzt.
- Dokumentierte Lizenzierung mit [REUSE 3.0](#).
- Die Aufteilung der Benutzerliste in der Teamverwaltung korrigiert.
- Verbesserte Leistung beim Erstellen und Speichern von Projekten.
- Unterstützung für gotext JSON-Dateien hinzugefügt.
- Unterstützung für i18next v4-Dateien hinzugefügt.
- Der Seitenumbruch in der API ist jetzt anpassbar.

[Alle Änderungen im Detail.](#)

4.4 Weblate 4.15

Veröffentlicht am 16. Dezember 2022.

- Unterstützung für das Durchsuchen von Änderungen für eine einzelne Zeichenkette hinzugefügt.
- Die Behandlung von Pluralen bei der automatischen Übersetzung aus anderen Komponenten korrigiert.
- Tastaturkürzel Alt+Eingabe hinzugefügt, um eine Zeichenkette als Übersetzungsvorschlag zu übermitteln.
- Unterstützung für Platzhalter im Fluent-Format hinzugefügt.
- Verbesserte Leistung des Übersetzungsspeichers.
- Automatische Generierung von Repoweb-Browsing-Links für bekannte Code-Hosting-Dienste.
- Verbesserte Leistung mehrerer Ansichten.
- Verbesserte Auflistung von Zeichenketten mit Pluralformen.
- Unterstützung für das Hinzufügen von benutzerdefiniertem Markup zum HTML-Header hinzugefügt.
- Die Erzeugung von MO-Dateien in der Erweiterung korrigiert, so dass nur noch übersetzte Dateien enthalten sind.
- Die Darstellung von Markierungen für reguläre Ausdrücke korrigiert.
- Verbessertes Prüfverhalten von Platzhaltern bei Pluralformen.
- Unterstützung für die Benennung von Übersetzungsdateien, die für Google Play geeignet sind, hinzugefügt.
- Unterstützung für Labels in der API hinzugefügt.
- Unterstützung für die Auswahl einer anderen E-Mail für Commits als für Benachrichtigungen hinzugefügt.
- Das Docker-Image aktiviert nicht mehr standardmäßig den Debugmodus.
- Die Glossarbegriffe werden basierend auf der Priorität der Glossarkomponente sortiert.
- Teamadministratoren hinzugefügt, die Mitglieder des Teams hinzufügen oder entfernen können.
- Popup-Bestätigung vor dem Löschen von Benutzern hinzugefügt.
- Erweiterung zur Anpassung der XML-Ausgabe hinzugefügt.

[Alle Änderungen im Detail.](#)

4.5 Weblate 4.14.2

Veröffentlicht am 5. November 2022.

- Unterstützung für das Entfernen von Einträgen aus dem Übersetzungsspeicher hinzugefügt.
- Verbesserte Analyse der Warnung bei doppelter Sprache.
- Verbesserte Genauigkeit bei der Prüfung auf aufeinanderfolgende doppelte Wörter.
- Verbesserte Skalierung des Versands vieler Benachrichtigungen.
- Verbesserte Behandlung des Zeichenkettenstatus bei der Übersetzung von Untertiteln.
- Deprecated insecure configuration of VCS service API keys via `_TOKEN/_USERNAME` configuration instead of `_CREDENTIALS` list.
- Die Verarbeitung einiger hochgeladener CSV-Dateien korrigiert.
- Verbesserte Handhabung von Leerzeichenänderungen in der Diff-Anzeige.
- Added automatic suggestions management link to management pages.
- Nachverfolgung der Entfernung/Lösung von Kommentaren im Verlauf.
- Die Wiederherstellung von Projektsicherungen mit verknüpften Komponenten korrigiert.
- Captcha-Eingabe bei erfolgloser Registrierung behoben.
- Verbesserte Sprachunterstützung in DeepL.
- Verbesserte Kompatibilität von Webhooks mit authentifizierten Repositories.
- Unterstützung für Python 3.11 hinzugefügt.

[Alle Änderungen im Detail.](#)

4.6 Weblate 4.14.1

Veröffentlicht am 15. September 2022.

- Die Erstellung von Projekt-Backups in einigen Situationen korrigiert.
- Verbesserte Fehlerberichterstattung beim Hochladen von Dateien.
- Fetch all user verified e-mails from GitHub during authentication.
- Avoid matching glossary terms on context or keys.
- Benachrichtigungen für das Entfernen von Zeichenketten hinzugefügt.
- Verbesserte Verwaltung von nicht übersetzbaren Begriffen im Glossar.
- Auflistung der Teammitgliederanzahl auf der Teamverwaltungsseite.
- Benutzeroberfläche für die Gruppenverwaltung hinzugefügt.
- Überprüfungsstatus immer anzeigen, wenn Überprüfungen aktiviert sind.
- Unterstützung für die Suche in der Einheiten-API hinzugefügt.
- Fixed progress bar display for read-only strings in the review workflow.
- Verbesserte Interpunktionsprüfung für Burmesisch.
- Fixed garbage collecting of metrics data.

[Alle Änderungen im Detail.](#)

4.7 Weblate 4.14

Veröffentlicht am 22. August 2022.

- Nachverfolgung der Änderungen an Erweiterungen in einem Verlauf.
- Das Parsen der Übersetzung aus Windows RC-, HTML- und Textdateien korrigiert.
- Extended language code style configuration options.
- Unterstützung für Pluralformen hinzugefügt, die in den letzten CLDR-Versionen aktualisiert wurden.
- Geringerer Speicherverbrauch beim Aktualisieren von Komponenten mit vielen Übersetzungen.
- Unterstützung für die Übersetzungsdomäne im SAP Translation Hub hinzugefügt.
- Absolute Links in Quelltexten von Ausgangszeichenketten ermöglicht.
- Improved operation behind some reverse proxies.
- Erweiterte API zur Einbeziehung des Übersetzungsspeichers.
- Übersetzungsabläufe für Dokumente verbessert.
- Die Zuverlässigkeit der Übersetzung von HTML- und Textdateien verbessert.
- Unterstützung für Backups auf Projektebene hinzugefügt.
- Verbesserte Leistung und Speichernutzung vom Übersetzungsspeicher.

[Alle Änderungen im Detail.](#)

4.8 Weblate 4.13.1

Veröffentlicht am 1. Juli 2022.

- Tracking-Vorschläge im Verlauf behoben.
- Das Parsen von Reverse-Proxy-Informationen von Cloudflare korrigiert.
- Make parse error lock a component from translating.
- Die Konfiguration der Zwischendatei in der Discovery-Erweiterung korrigiert.
- Das Verhalten von DeepL bei Übersetzungen mit Platzhaltern korrigiert.
- Nichtübersetzen von Zeichenketten über die API behoben.
- Unterstützung für das Entfernen von Benutzern aus einer Gruppe über die API hinzugefügt.
- Das Audit-Protokoll für Einladungs-E-Mails für Benutzer korrigiert.
- Markierungsnamen für Java-Formatierungszeichenketten korrigiert.

[Alle Änderungen im Detail.](#)

4.9 Weblate 4.13

Veröffentlicht am 15. Juni 2022.

- Changed behavior of updating language names.
- Seitenumbruch zur Projektliste hinzugefügt.
- API for creating new units now returns information about newly created unit.
- Die Komponentenerkennung unterstützt nun die Konfiguration einer Zwischensprache.

- Added fixed encoding variants to CSV formats.
- Changed handling of context and location for some formats to better fit underlying implementation.
- Unterstützung für das ResourceDictionary-Format hinzugefügt.
- Improved progress bar colors for color blind.
- Bereinigung von Varianten beim Entfernen einer Zeichenkette behoben.
- Kompatibilität mit Django 4.1.
- Unterstützung für das Speichern von XML-Elementen mit Escapezeichen in XLIFF hinzugefügt.
- Verbesserte Formatierung von Fehlern bei Platzhalterprüfungen.
- Redirect /.well-known/change-password to /accounts/password/.
- Maschinelle Übersetzungsdienste sind jetzt pro Projekt konfigurierbar.
- Added separate permission for resolving comments and grant it to the *Review strings* role.
- Unterstützung für die Speicherung alternativer Übersetzungen in der CSV-Datei hinzugefügt.
- The placeholders check can now be case-insensitive as well.

[Alle Änderungen im Detail.](#)

4.10 Weblate 4.12.2

Veröffentlicht am 11. Mai 2022.

- Die Wiederherstellung des Projektübersetzungsspeichers für einige Komponenten korrigiert.
- Fehler bei Sortierung von Komponenten nach nicht übersetzten Zeichenketten behoben.
- Möglicher Verlust von Übersetzungen beim Hinzufügen einer neuen Sprache behoben.
- Ensure Weblate SSH key is generated during migrations.

[Alle Änderungen im Detail.](#)

4.11 Weblate 4.12.1

Veröffentlicht am 29. April 2022.

- Pull-Request-Nachrichtentitel korrigiert.
- Verbesserte Syntaxfehlerbehandlung im Fluent-Format.
- Die Anzeige vom Profilbild in Benachrichtigungs-E-Mails korrigiert.
- Unterstützung für Web-Monetarisierung hinzugefügt.
- Löschung von alten Ausgangszeichenketten während Entfernung von Übersetzungen behoben.

[Alle Änderungen im Detail.](#)

4.12 Weblate 4.12

Veröffentlicht am 20. April 2022.

- Unterstützung für Amharisch in *Nicht übereinstimmender Punkt* hinzugefügt.
- Unterstützung für Birmanisch in *Nicht übereinstimmendes Fragezeichen* hinzugefügt.
- Erweiterte Optionen der Erweiterung: *Erzeugung von Pseudolokalen*.
- Markierung `ignore-all-checks` hinzugefügt, um alle Qualitätsprüfungen für eine Zeichenkette zu ignorieren.
- Die Erweiterung *Erzeugung von Pseudolokalen* meiden, um fehlende Qualitätsprüfungen auszulösen.
- Unterstützung für *Gitea-Pull-Requests* hinzugefügt.
- Sprachcode im Linux-Stil zu *Stil des Sprachcodes* hinzugefügt.
- Unterstützung für den Wiederaufbau von Projektübersetzungsspeichern wurde hinzugefügt.
- Verbesserte API für die Erstellung von Komponenten aus einer Datei.
- Hinzufügen von Schaltflächen zum Kopieren und Klonen zu anderen Übersetzungen.
- Konfigurierbarkeit der Nachricht für Merge Request auf Komponentenebene.
- Improved maximal length restriction behavior with XML tags.
- Das Laden von Fluent-Dateien mit zusätzlichen Kommentaren korrigiert.

[Alle Änderungen im Detail.](#)

4.13 Weblate 4.11.2

Veröffentlicht am 4. März 2022.

- Beschädigte MO-Dateien in der Binärversion behoben.

[Alle Änderungen im Detail.](#)

4.14 Weblate 4.11.1

Veröffentlicht am 4. März 2022.

- Fixed missing sanitizing of arguments to Git and Mercurial - CVE-2022-23915, see [GHSA-3872-f48p-pxqj](#) for more details.
- Das Laden von fragwürdigen Zeichenketten aus CSV-Dateien korrigiert.
- Unterstützung für die Erstellung von Teams über die API hinzugefügt.
- Die Anzeige von Vorschlägen für Benutzererwähnungen wurde korrigiert.
- The project tokens access can now be customized.

[Alle Änderungen im Detail.](#)

4.15 Weblate 4.11

Veröffentlicht am 25. Februar 2022.

- Fixes stored XSS - CVE-2022-24710, see [GHSA-6jp6-9rf9-gc66](#) for more details.
- Installation von Erweiterungen über die API korrigiert.
- Umbenennung von *Zeichenketten mit Handlungsbedarf* in *Unvollständige Zeichenketten*.
- Fehlalarme von *ICU MessageFormat-Syntax* behoben.
- Indicate lock and contributor agreement on other occurrences listing.
- Fixed updating PO files with obsolete strings or missing plurals.
- Die Kompatibilität der Squash-Erweiterung mit Gerrit verbessert.
- Automatically initialize user languages based on the *Accept-Language* header.
- Verbesserte Fehlerbehandlung beim Entfernen von Zeichenketten.
- Weblate erfordert jetzt Python 3.7 oder neuer.
- Einige Schreiboperationen mit der Projekt-Token-Authentifizierung behoben.
- Die Nachverfolgung des Zeichenkettenstatus bei Änderungen der Zeichenkette im Repository korrigiert.
- Zeichenkettenveränderungen im Repository verfolgen.
- Sticky header on translations listing to improve navigation.
- Nicht übersetzte Zeichenketten in *Java properties* behoben.
- Fixed Git operation with non-ascii branch names.
- Neue Erweiterung *Übersetzung mit Quelle vorbelegen*.
- *Git-Strategie Zusammenführen ohne Fast-Forward* hinzugefügt.
- Erweiterungstrigger für neu hinzugefügte Zeichenketten *Automatische Übersetzung* repariert.
- Verbesserte Interpunktionsprüfung für Burmesisch.
- Unterstützung zur Festlegung benutzerdefinierter Teams auf Projektebene, um Benutzerzugriffe zu gewähren, hinzugefügt, siehe *Verwaltung der projektspezifischen Zugriffssteuerung*.
- Links zur Dokumentation bei Warnungen hinzugefügt.
- Docker-Container aktiviert bei Bedarf automatisch TLS/SSL für ausgehende E-Mails.
- Unterstützung für die Suche nach geklärten Kommentaren ergänzt.
- Unterstützung für borgbackup 1.2 hinzugefügt.
- Die Verwendung von *Automatisch übersetzt* korrigiert.

[Alle Änderungen im Detail.](#)

4.16 Weblate 4.10.1

Veröffentlicht am 22. Dezember 2021.

- Dokumentierte Änderungen, die durch das Upgrade auf Django 4.0 eingeführt wurden.
- Die Anzeige von *Automatisch übersetzt* korrigiert.
- Die API-Anzeige einer Branch in Komponenten mit einem gemeinsamen Repository korrigiert.
- Verbesserte Analyse der fehlgeschlagenen Push-Warnung.
- Das manuelle Bearbeiten der Seite beim Durchsuchen von Änderungen behoben.

- Die Genauigkeit von *Kashida-Buchstabe verwendet* verbessert.
- Der Weblate-Docker-Container verwendet jetzt Python 3.10.

[Alle Änderungen im Detail.](#)

4.17 Weblate 4.10

Veröffentlicht am 16. Dezember 2021.

- Unterstützung für Formalitäten und Platzhalter mit DeepL hinzugefügt.
- Massенbearbeitung sowie Suchen und Ersetzen sind jetzt auf Projekt- und Sprachebene verfügbar.
- Filterung beim Suchen und Ersetzen hinzugefügt.
- Fixed: „Perform automatic translation“ privilege is no longer part of the *Languages* group.
- „Perform automatic translation“ is in the *Administration* and the new *Automatic translation* group.
- Das Erzeugen von XLSX-Dateien mit Sonderzeichen korrigiert.
- Dem GitHub-Authentifizierungs-Backend die Möglichkeit hinzugefügt, zu prüfen, ob der Benutzer zu einer bestimmten GitHub-Organisation oder einem Team gehört.
- Verbesserte Rückmeldung über ungültige an die API übergebene Parameter.
- Unterstützung für projektspezifische Zugriffstoken für die API hinzugefügt.
- Die Entfernung von Zeichenketten in einigen Fällen korrigiert.
- Die Übersetzung neu hinzugefügter Zeichenketten korrigiert.
- Kennzeichnung automatisch übersetzter Zeichenketten, um ihre Filterung zu erleichtern.

[Alle Änderungen im Detail.](#)

4.18 Weblate 4.9.1

Veröffentlicht am 19. November 2021.

- Fixed upload of monolingual files after changing template.
- Verbesserte Handhabung von Leerzeichen in Markierungen.
- Unterstützung für die Filterung in der Download-API hinzugefügt.
- Die Statistikanzeige beim Hinzufügen neuer Übersetzungen korrigiert.
- Mitigate issues with GitHub SSH key change.

[Alle Änderungen im Detail.](#)

4.19 Weblate 4.9

Veröffentlicht am 10. November 2021.

- Provide more details for events in history.
- Verbesserte Darstellung des Verlaufs.
- Verbesserte Leistung der Übersetzungsseiten.
- Unterstützung für die Einschränkung des Downloads von Übersetzungsdateien hinzugefügt.
- The `safe-html` can now understand Markdown when used with `md-text`.

- The `max-length` tag now ignores XML markup when used with `xml-text`.
- Fixed dimensions of rendered texts in *Maximaler Umfang der Übersetzung*.
- Lowered app store title length to 30 to assist with upcoming Google policy changes.
- Unterstützung für die Anpassung des SSH-Aufrufs über `SSH_EXTRA_ARGS` hinzugefügt.
- Qualitätsprüfungen für ICU-MessageFormat hinzugefügt.
- Verbesserte Behandlung von Fehlerbedingungen in maschinellen Übersetzungs-Backends.
- Highlight unusual whitespace characters in the strings.
- Option hinzugefügt, um beim Bearbeiten auf der übersetzten Zeichenkette zu bleiben.
- Unterstützung für die Anpassung des Borg-Aufrufs über `BORG_EXTRA_ARGS` hinzugefügt.
- Die Erzeugung von MO-Dateien für einsprachige Übersetzungen korrigiert.
- Added API endpoint to download all component translations as a ZIP file.
- Unterstützung für Python 3.10 hinzugefügt.
- Unterstützung für den erneuten Versand von E-Mail-Einladungen über die Verwaltungsschnittstelle hinzugefügt.

[Alle Änderungen im Detail.](#)

4.20 Weblate 4.8.1

Veröffentlicht am 10. September 2021.

- Das Entfernen von Benutzern in der Django-Adminoberfläche behoben.
- Ausführlichere Dokumentation von Erweiterungsparametern.
- JavaScript-Fehler im Glossar behoben.
- Begrenzung der Anzahl von Übereinstimmungen bei der Konsistenzprüfung.
- Verbesserung der Behandlung von Platzhaltern in maschinellen Übersetzungen.
- Die Erstellung von Erweiterungen über die API behoben.
- Die Einstellung `PRIVACY_URL` hinzugefügt, um einen Link zur Datenschutzbestimmungen in die Fußzeile einzufügen.
- Hide member e-mail addresses from project admins.
- Improved gettext PO merging in case of conflicts.
- Verbesserte Hervorhebung des Glossars.
- Improved `safe-html` flag behavior with XML checks.
- Commit-Nachrichten für verknüpfte Komponenten korrigiert.

[Alle Änderungen im Detail.](#)

4.21 Weblate 4.8

Veröffentlicht am 21. August 2021.

- Unterstützung für das Apple-Stringsdict-Format hinzugefügt.
- Der Operator für die exakte Suche unterscheidet jetzt bei PostgreSQL zwischen Groß- und Kleinschreibung.
- Das Speichern von Glossarerklärungen in einigen Fällen korrigiert.
- Verbesserung der Dokumentation.
- Leistungsverbesserungen.
- Die Kompatibilität der Squash-Erweiterung mit Gerrit verbessert.
- Das Hinzufügen von Zeichenketten zu einsprachigen Glossarkomponenten korrigiert.
- Verbesserte Leistung bei der Handhabung von Varianten.
- Das gelegentliche Überspringen des Parsens von Upstream-Änderungen der Squash-Erweiterung behoben.
- Dateierweiterung für Downloads beibehalten.
- Unterstützung für das Fluent-Format hinzugefügt.
- Unterstützung für die Verwendung von Tabulatoren zum Einrücken von JSON-Formaten hinzugefügt.

[Alle Änderungen im Detail.](#)

4.22 Weblate 4.7.2

Veröffentlicht am 15. Juli 2021.

- Unterstützung von mehr Sprachaliasen, die für ein Projekt konfiguriert werden können.
- Die Validierung der Suchbegriffe in der API korrigiert.
- Git-Exporter-URLs nach einem Domainwechsel korrigiert.
- Bereinigungsvorgang der Erweiterung für Windows RC-Dateien korrigiert.
- Möglicher Absturz bei der XLIFF-Aktualisierung behoben.

[Alle Änderungen im Detail.](#)

4.23 Weblate 4.7.1

Veröffentlicht am 30. Juni 2021.

- Verbessertes Popup beim Hinzufügen von Begriffen zum Glossar.
- Unterstützung für den maschinellen Übersetzungsdienst LibreTranslate hinzugefügt.
- Eine Ratenbegrenzung für die Erstellung neuer Projekte hinzugefügt.
- Die Leistung von Dateiaktualisierungen verbessert.

[Alle Änderungen im Detail.](#)

4.24 Weblate 4.7

Veröffentlicht am 17. Juni 2021.

- Verbesserte Zustandsprüfung der Konfiguration.
- Unterstützung für `object-pascal-format` in gettext PO hinzugefügt, siehe *Object Pascal-Format*.
- Umbenennung von *Benachbarte Schlüssel* in *Ähnliche Schlüssel*, um den Zweck besser zu beschreiben.
- Unterstützung für *mil8n lang Dateien* hinzugefügt.
- Verbesserte Integration der SAML-Authentifizierung.
- Die Integration von *Gerrit* korrigiert, um Ausnahmefälle besser zu handhaben.
- Weblate benötigt jetzt Django 3.2.
- Das Einladen von Benutzern bei deaktivierter E-Mail-Authentifizierung behoben.
- Verbesserte Sprachdefinitionen.
- Unterstützung für das Sperren von Benutzern für Beiträge zu einem Projekt hinzugefügt.
- Die automatische Erstellung von Glossarsprachen korrigiert.
- Erweiterte Dokumentation zu Erweiterungen.
- Leistungsverbesserungen für Komponenten mit verknüpften Repositories.
- Unterstützung für kostenlose DeepL-API hinzugefügt.
- Die Benutzerverwaltung benötigt keine Django-Adminoberfläche mehr.

[Alle Änderungen im Detail.](#)

4.25 Weblate 4.6.2

Veröffentlicht am 8. Mai 2021.

- Absturz nach dem Verschieben einer gemeinsamen Komponente zwischen Projekten behoben.
- Das Hinzufügen neuer Zeichenketten zu leeren Eigenschaftendateien korrigiert.
- Die Ausrichtung der Kopiersymbole in Sprachen mit linksläufiger Schreibrichtung korrigiert.
- Erweiterte Zeichenkettenstatistik auf der Information-Reiterkarte.
- Die Behandlung von in Git ignorierten Übersetzungsdateien korrigiert.
- Verbesserte Leistung der Metriken.
- Möglicher Fehler beim Speichern von Glossaren behoben.
- Das Verhalten der Konsistenzprüfung bei Sprachen mit unterschiedlichen Pluralregeln korrigiert.

[Alle Änderungen im Detail.](#)

4.26 Weblate 4.6.1

Veröffentlicht am 2. Mai 2021.

- Veralteten Spamschutz-Code entfernt.
- Verbessern Sie die Genauigkeit der Pluralprüfung von Ausgangszeichenketten.
- Aktualisierung der Liste der Benutzeroberflächensprachen in Docker.
- Verbesserte Fehlermeldungen beim Erstellen von Pull Requests.
- Die Erstellung von Pull Requests auf Pagure behoben.
- Das Auslösen von automatisch installierten Erweiterungen behoben.
- Mögliche Caching-Probleme beim Upgrade behoben.
- Das Hinzufügen neuer Einheiten zu einsprachigen Übersetzungen mittels Upload behoben.

[Alle Änderungen im Detail.](#)

4.27 Weblate 4.6

Veröffentlicht am 19. April 2021.

- Der Verwaltungsbefehl `auto_translate` hat jetzt einen Parameter zur Angabe des Übersetzungsmodus.
- Unterstützung für *Textdateien* hinzugefügt.
- Trends und Metriken für alle Objekte hinzugefügt.
- Unterstützung für direktes Kopieren von Text aus Zweitsprachen hinzugefügt.
- Datumsfilterung beim Durchsuchen von Änderungen hinzugefügt.
- Verbesserte Aktivitätsdiagramme.
- Absender für Kontaktformular-E-Mails können jetzt konfiguriert werden.
- Verbesserte Parametervalidierung in der API der Komponentenerstellung.
- The rate limiting no longer applies to superusers.
- Die Leistung und Zuverlässigkeit der Erweiterung für die automatische Übersetzung verbessert.
- Die Ratenbegrenzung kann nun im Docker-Container angepasst werden.
- API zum Erstellen von Komponenten verwendet jetzt automatisch *Weblate-interne URLs*.
- Vereinfachte Statusanzeige beim Auflisten von Zeichenketten.
- Passwort-Hashing verwendet jetzt standardmäßig Argon2.
- Vereinfachte Fortschrittsbalken, die den Übersetzungsstatus anzeigen.
- *Fehlende Sprachen hinzufügen* umbenannt, um den Zweck zu verdeutlichen.
- Das Speichern des Zeichenkettenstatus in XLIFF korrigiert.
- Sprachübergreifende Suche hinzugefügt.
- Erstmalige Unterstützung für *Horizontale Skalierung* der Docker-Bereitstellung.

[Alle Änderungen im Detail.](#)

4.28 Weblate 4.5.3

Veröffentlicht am 1. April 2021.

- Metriksammlung korrigiert.
- Möglicher Absturz beim Hinzufügen von Zeichenketten behoben.
- Verbesserte Beispiele für Suchanfragen.
- Möglicher Verlust von neu hinzugefügten Zeichenketten beim Ersetzen des Hochladens behoben.

4.29 Weblate 4.5.2

Veröffentlicht am 26. März 2021.

- Konfigurierbarer Zeitplan für die automatische Übersetzung.
- Lua-Format-Prüfung hinzugefügt.
- Ignore format strings in the *Aufeinanderfolgende doppelte Wörter* check.
- Erlaubt das Hochladen von Bildschirmfotos von einer Übersetzungsseite.
- Erzwungene Dateisynchronisation zur Paketquellwartung hinzugefügt.
- Fixed automatic suggestions for languages with a longer code.
- Verbesserte Leistung beim Hinzufügen neuer Zeichenketten.
- Mehrere Fehlerbehebungen bei den Qualitätsprüfungen.
- Mehrere Leistungsverbesserungen.
- Integration mit *Entdecken Sie Weblate* hinzugefügt.
- Fixed checks behavior with read-only strings.

[Alle Änderungen im Detail.](#)

4.30 Weblate 4.5.1

Veröffentlicht am 5. März 2021.

- Die Bearbeitung von Glossar-Markierungen in einigen Ausnahmefällen korrigiert.
- Extend metrics usage to improve performance of several pages.
- Store correct source language in TMX files.
- Bessere Handhabung beim Hochladen von einsprachigen PO über die API.
- Das Verhalten von Warnmeldungen bei Glossarkomponenten verbessert.
- Verbesserte Markdown-Link-Prüfungen.
- Indicate glossary and source language in breadcrumbs.
- Komponentenliste mit Seitenumbruch bei großen Projekten.
- Verbesserte Leistung beim Entfernen von Übersetzungen, Komponenten oder Projekten.
- Improved bulk edit performance.
- Fixed preserving „Needs editing“ and „Approved“ states for ODF files.
- Verbesserte Benutzeroberfläche zum Anpassen der Downloads von Übersetzungsdateien

[Alle Änderungen im Detail.](#)

4.31 Weblate 4.5

Veröffentlicht am 19. Februar 2021.

- Unterstützung für das in gettext PO verwendete `lua-format` hinzugefügt.
- Unterstützung für gemeinsame Nutzung einer Komponente zwischen Projekten hinzugefügt.
- Fixed multiple unnamed variables check behavior with multiple format flags.
- Dropped mailing list field on the project in favor of generic instructions for translators.
- Erweiterung zur Erzeugung von Pseudolokalen hinzugefügt.
- Unterstützung für TermBase eXchange-Dateien hinzugefügt.
- Unterstützung für die manuelle Definition von Zeichenkette-Varianten mit Hilfe einer Markierung hinzugefügt.
- Verbesserte Leistung der Konsistenzprüfungen.
- Verbesserte Leistung des Übersetzungsspeichers für lange Zeichenketten.
- Unterstützung für die Suche in Erklärungen hinzugefügt.
- Strings can now be added and removed in bilingual formats as well.
- Extend list of supported languages in Amazon Translate machine translation.
- Automatische Aktivierung von Prüfungen der Eigenschaften des Java MessageFormat.
- Eine neue Upload-Methode hinzugefügt, um neue Zeichenketten zu einer Übersetzung hinzuzufügen.
- Eine einfache Benutzeroberfläche zum Durchsuchen von Übersetzungen hinzugefügt.
- Glossare werden jetzt als reguläre Komponenten gespeichert.
- Dropped specific API for glossaries as component API is used now.
- Added simplified interface to toggle some of the flags.
- Unterstützung für nicht übersetzbare oder unzulässige Begriffe im Glossar hinzugefügt.
- Unterstützung für die Definition von Terminologie in einem Glossar hinzugefügt.
- Der Umschalter für die Leserichtung wurde verschoben, um mehr Platz für die visuelle Tastatur zu erhalten.
- Option zum automatischen Beobachten von Projekten hinzugefügt, zu denen Benutzer beigetragen haben.
- Added check whether translation matches the glossary.
- Unterstützung für die Anpassung der Navigationstextfarbe hinzugefügt.

[Alle Änderungen im Detail.](#)

4.32 Weblate 4.4.2

Veröffentlicht am 14. Januar 2021.

- Fixed corruption of one distributed MO file.

4.33 Weblate 4.4.1

Veröffentlicht am 13. Januar 2021.

- Fixed reverting plural changes.
- Die Anzeige der Hilfe für Projekteinstellungen korrigiert.
- Verbesserte Benutzerverwaltung.
- Verbesserte Handhabung des Kontexts in einsprachigen PO-Dateien.
- Das Verhalten der Bereinigungsvorgang-Erweiterung bei den Formaten HTML, ODF, IDML und Windows RC korrigiert.
- Das Parsen des Speicherorts aus CSV-Dateien korrigiert.
- Use content compression for file downloads.
- Improved user experience on importing from ZIP file.
- Die Erkennung des Dateiformats beim Hochladen verbessert.
- Vermeidung doppelter Pull Requests auf Pagure.
- Verbesserte Leistung bei der Anzeige von Ghost-Übersetzungen.
- Reimplemented translation editor to use native browser textarea.
- Fehlerhafte Funktion der Erweiterung für den Bereinigungsvorgang beim Hinzufügen von neuen Zeichenketten behoben.
- API für Erweiterungen hinzugefügt.

[Alle Änderungen im Detail.](#)

4.34 Weblate 4.4

Veröffentlicht am 15. Dezember 2020.

- Verbesserte Validierung beim Erstellen einer Komponente.
- Weblate benötigt jetzt Django 3.1.
- Unterstützung für die Anpassung des Erscheinungsbilds in der Verwaltungsoberfläche hinzugefügt.
- Fixed read-only state handling in bulk edit.
- Verbesserte CodeMirror-Integration.
- Erweiterung hinzugefügt, um leere Zeichenketten aus Übersetzungsdateien zu entfernen.
- Der CodeMirror-Editor wird jetzt für Übersetzungen verwendet.
- Syntaxhervorhebung im Übersetzungseditor für XML, HTML, Markdown und reStructuredText.
- Platzhalter im Übersetzungseditor hervorheben.
- Verbesserte Unterstützung für nicht standardisierte Sprachcodes.
- Warnung bei Verwendung mehrdeutiger Sprachcodes hinzugefügt.
- Dem Benutzer wird nun eine gefilterte Liste von Sprachen angezeigt, wenn er eine neue Übersetzung hinzufügt.
- Erweiterte Suchmöglichkeiten für Änderungen im Verlauf.
- Improved billing detail pages and Libre hosting workflow.
- Erweiterte Übersetzungsstatistik-API.
- Verbesserte Reiterkarte „Andere Übersetzungen“ beim Übersetzen.

- Added tasks API.
- Die Leistung des Dateiuploads verbessert.
- Improved display of user defined special characters.
- Verbesserte Leistung der automatischen Übersetzung.
- Several minor improvements in the user interface.
- Improved naming of ZIP downloads.
- Option zum Erhalten von Benachrichtigungen über nicht beobachtete Projekte hinzugefügt.

[Alle Änderungen im Detail.](#)

4.35 Weblate 4.3.2

Veröffentlicht am 4. November 2020.

- Absturz bei bestimmten Komponentendateimasken behoben.
- Die Genauigkeit der Prüfung auf aufeinanderfolgende doppelte Wörter verbessert.
- Unterstützung für Pagure-Pull-Requests hinzugefügt.
- Verbesserte Fehlermeldungen bei fehlgeschlagenen Registrierungen.
- Reverted rendering developer comments as Markdown.
- Simplified setup of Git repositories with different default branch than „master“.
- Newly created internal repositories now use main as the default branch.
- Reduced false positives rate of unchanged translation while translating reStructuredText.
- Fixed CodeMirror display issues in some situations.
- Renamed Template group to „Sources“ to clarify its meaning.
- GitLab-Pull-Requests für Repositories mit längeren Pfaden korrigiert.

[Alle Änderungen im Detail.](#)

4.36 Weblate 4.3.1

Veröffentlicht am 21. Oktober 2020.

- Verbesserte Leistung der automatischen Übersetzung.
- Fixed session expiry for authenticated users.
- Unterstützung für das Ausblenden von Versionsinformationen hinzufügen.
- Verbesserte Hooks-Kompatibilität mit Bitbucket-Server.
- Die Leistung bei Aktualisierungen des Übersetzungsspeichers verbessert.
- Reduced memory usage.
- Verbesserte Leistung der Matrixansicht.
- Bestätigung vor dem Entfernen eines Benutzers aus einem Projekt hinzugefügt.

[Alle Änderungen im Detail.](#)

4.37 Weblate 4.3

Veröffentlicht am 15. Oktober 2020.

- Include user stats in the API.
- Fixed component ordering on paginated pages.
- Define source language for a glossary.
- Überarbeitete Unterstützung für GitHub- und GitLab-Pull-Requests.
- Behobene Statistikzählungen nach dem Entfernen eines Übersetzungsvorschlags.
- Erweitertes öffentliches Benutzerprofil.
- Fixed configuration of enforced checks.
- Verbesserung der Dokumentation über integrierte Backups.
- Moved source language attribute from project to a component.
- „Vue I18n“-Formatierungsprüfung hinzugefügt.
- Die allgemeine Platzhalterprüfung unterstützt jetzt reguläre Ausdrücke.
- Verbessertes Aussehen des Matrixmodus.
- Suchmaschinen werden jetzt als automatische Vorschläge bezeichnet.
- Unterstützung für die Interaktion mit mehreren GitLab- oder GitHub-Instanzen hinzugefügt.
- Extended API to cover project updates, unit updates and removals and glossaries.
- Unit API now properly handles plural strings.
- Component creation can now handle ZIP file or document upload.
- Consolidated API response status codes.
- Unterstützung von Markdown in der Mitwirkungsvereinbarung.
- Verbesserte Verfolgung von Ausgangszeichenketten.
- Verbesserte Kompatibilität mit JSON-, YAML- und CSV-Formaten.
- Unterstützung für das Entfernen von Zeichenketten hinzugefügt.
- Verbesserte Leistung beim Herunterladen von Dateien.
- Verbesserte Ansicht der Repository-Verwaltung.
- Automatisches Aktivieren des Java-Formats für Android.
- Unterstützung für lokalisierte Bildschirmfotos hinzugefügt.
- Unterstützung für Python 3.9 hinzugefügt.
- Die Übersetzung von HTML-Dateien wurde unter bestimmten Bedingungen korrigiert.

[Alle Änderungen im Detail.](#)

4.38 Weblate 4.2.2

Veröffentlicht am 2. September 2020.

- Abgleich von Ausgangszeichenketten für JSON-Formate korrigiert.
- Die Umleitung der Anmeldung für einige Authentifizierungskonfigurationen korrigiert.
- LDAP-Authentifizierung mit Gruppensynchronisierung korrigiert.
- Absturz bei der Meldung des automatischen Übersetzungsfortschritts behoben.
- Git-Commit-Zusammenfassungen bei aktivierten Endzeilen korrigiert.
- Die Erstellung lokaler VCS-Komponenten über die API korrigiert.

4.39 Weblate 4.2.1

Veröffentlicht am 21. August 2020.

- Fixed saving plurals for some locales in Android resources.
- Absturz in der Erweiterung für den Bereinigungsverfahren für einige XLIFF-Dateien behoben.
- Allow setting up localization CDN in Docker image.

4.40 Weblate 4.2

Veröffentlicht am 18. August 2020.

- Improved user pages and added listing of users.
- Die Unterstützung für die Migration von 3.x-Versionen wurde eingestellt, die Migration erfolgt über 4.1 oder 4.0.
- Added exports into several monolingual formats.
- Verbesserte Aktivitätsdiagramme.
- Number of displayed nearby strings can be configured.
- Unterstützung für das Sperren von Komponenten, bei denen Repository-Fehler auftreten, hinzugefügt.
- Simplified main navigation (replaced buttons with icons).
- Improved language code handling in Google Translate integration.
- Die Git-Squash-Erweiterung kann `Co-authored-by:`-Endzeilen erzeugen.
- Improved query search parser.
- Improved user feedback from format strings checks.
- Improved performance of bulk state changes.
- Added compatibility redirects after project or component renaming.
- Added notifications for strings approval, component locking and license change.
- Unterstützung für ModernMT hinzugefügt.
- Allow to avoid overwriting approved translations on file upload.
- Die Unterstützung für einige Kompatibilitäts-Weiterleitungs-URLs eingestellt.
- Prüfung für ECMAScript-Buchstabenvorlagen hinzugefügt.
- Option zum Beobachten einer Komponente hinzugefügt.

- Removed leading dot from JSON unit keys.
- Die separate Celery-Warteschlange für den Übersetzungsspeicher wurde entfernt.
- Erlaubt das Übersetzen aller Komponenten einer Sprache auf einmal.
- Allow to configure Content-Security-Policy HTTP headers.
- Added support for aliasing languages at project level.
- Neue Erweiterung zur Hilfe bei der HTML- oder JavaScript-Lokalisierung, siehe [JavaScript-Lokalisierung CDN](#).
- The Weblate domain is now configured in the settings, see [SITE_DOMAIN](#).
- Unterstützung für die Suche nach Komponente und Projekt hinzufügen.

4.41 Weblate 4.1.1

Veröffentlicht am 19. Juni 2020.

- Änderung der Autofix- oder Erweiterungs-Konfiguration in Docker behoben.
- Fixed possible crash in „About“ page.
- Improved installation of byte-compiled locale files.
- Fixed adding words to glossary.
- Fixed keyboard shortcuts for machinery.
- Removed debugging output causing discarding log events in some setups.
- Fixed lock indication on project listing.
- Fixed listing GPG keys in some setups.
- Added option for which DeepL API version to use.
- Added support for acting as SAML Service Provider, see [SAML-Authentifizierung](#).

4.42 Weblate 4.1

Veröffentlicht am 15. Juni 2020.

- Added support for creating new translations with included country code.
- Unterstützung für die Suche nach Ausgangszeichenketten mit Bildschirmfoto hinzugefügt.
- Extended info available in the stats insights.
- Improved search editing on „Translate“ pages.
- Improve handling of concurrent repository updates.
- Include source language in project creation form.
- Include changes count in credits.
- Fixed UI language selection in some cases.
- Allow to whitelist registration methods with registrations closed.
- Improved lookup of related terms in glossary.
- Improved translation memory matches.
- Group same machinery results.
- Direkten Link zum Bearbeiten des Bildschirmfotos von der Übersetzungsseite hinzugefügt.

- Improved removal confirmation dialog.
- Include templates in ZIP download.
- Add support for Markdown and notification configuration in announcements.
- Extended details in check listings.
- Unterstützung für neue Dateiformate hinzugefügt: *Laravel PHP-Zeichenketten*, *HTML-Dateien*, *OpenDocument-Format*, *IDML-Format*, *Windows RC-Dateien*, *INI-Übersetzungen*, *Inno Setup INI-Übersetzungen*, *GWT-Eigenschaften*, *go-i18n-JSON-Dateien*, *ARB-Datei*.
- Consistently use dismissed as state of dismissed checks.
- Unterstützung für die Aktivierung der Konfiguration von Standarderweiterungen hinzugefügt.
- Fixed editor keyboard shortcut to dismiss checks.
- Improved machine translation of strings with placeholders.
- Show ghost translation for user languages to ease starting them.
- Improved language code parsing.
- Show translations in user language first in the list.
- Renamed shapings to more generic name variants.
- Added new quality checks: *Mehrere unbenannte Variablen*, *Lange nicht übersetzt*, *Aufeinanderfolgende doppelte Wörter*.
- Reintroduced support for wiping translation memory.
- Fixed option to ignore source checks.
- Added support for configuring different branch for pushing changes.
- API now reports rate limiting status in the HTTP headers.
- Unterstützung für Google Translate V3 API (Advanced) hinzugefügt.
- Added ability to restrict access on component level.
- Added support for whitespace and other special chars in translation flags, see *Anpassen des Verhaltens mit Markierungen*.
- Always show rendered text check if enabled.
- API now supports filtering of changes.
- Added support for sharing glossaries between projects.

4.43 Weblate 4.0.4

Veröffentlicht am 7. Mai 2020.

- Fixed testsuite execution on some Python 3.8 environments.
- Typo fixes in the documentation.
- Fixed creating components using API in some cases.
- Fixed JavaScript errors breaking mobile navigation.
- Fixed crash on displaying some checks.
- Die Auflistung der Bildschirmfotos korrigiert.
- Die monatlichen Zusammenfassungen-Benachrichtigungen korrigiert.
- Fixed intermediate translation behavior with units non existing in translation.

4.44 Weblate 4.0.3

Veröffentlicht am 2. Mai 2020.

- Fixed possible crash in reports.
- User mentions in comments are now case insensitive.
- Fixed PostgreSQL migration for non superusers.
- Fixed changing the repository URL while creating component.
- Fixed crash when upstream repository is gone.

4.45 Weblate 4.0.2

Veröffentlicht am 27. April 2020.

- Improved performance of translation stats.
- Improved performance of changing labels.
- Improved bulk edit performance.
- Improved translation memory performance.
- Fixed possible crash on component deletion.
- Die Anzeige von Übersetzungsänderungen in einigen Ausnahmefällen korrigiert.
- Improved warning about too long celery queue.
- Fixed possible false positives in the consistency check.
- Fixed deadlock when changing linked component repository.
- Included edit distance in changes listing and CSV and reports.
- Avoid false positives of punctuation spacing check for Canadian French.
- Fixed XLIFF export with placeholders.
- Fixed false positive with zero width check.
- Verbesserte Auswertung von Konfigurationsfehlern.
- Fixed bilingual source upload.
- Automatically detect supported languages for DeepL machine translation.
- Die Anzeige des Fortschrittsbalkens in einigen Ausnahmefällen korrigiert.
- Fixed some checks triggering on non translated strings.

4.46 Weblate 4.0.1

Veröffentlicht am 16. April 2020.

- Fixed package installation from PyPI.

4.47 Weblate 4.0

Veröffentlicht am 16. April 2020.

- Weblate now requires Python 3.6 or newer.
- Added management overview of component alerts.
- Added component alert for broken repository browser URLs.
- Improved sign in and registration pages.
- Project access control and workflow configuration integrated to project settings.
- Added check and highlighter for i18next interpolation and nesting.
- Added check and highlighter for percent placeholders.
- Vorschläge für fehlgeschlagene Prüfungen anzeigen.
- Record source string changes in history.
- Upgraded Microsoft Translator to version 3 API.
- Reimplemented translation memory backend.
- Added support for several `is:` lookups in *Suche*.
- Allow to make *Unveränderte Übersetzung* avoid internal blacklist.
- Improved comments extraction from monolingual po files.
- Renamed whiteboard messages to announcements.
- Fixed occasional problems with registration mails.
- Die LINGUAS-Update-Erweiterung verbessert, um mehr Syntaxvarianten zu behandeln.
- Fixed editing monolingual XLIFF source file.
- Unterstützung für exakte Übereinstimmung in *Suche* hinzugefügt.
- Erweiterte API zur Abdeckung von Bildschirmfotos, Benutzern, Gruppen, Komponentenlisten und erweiterter Projekterstellung.
- Add support for source upload on bilingual translations.
- Unterstützung für die Zwischensprachdatei von Entwicklern hinzugefügt.
- Added support for source strings review.
- Extended download options for platform wide translation memory.

4.48 Weblate 3.x-Reihe

4.48.1 Weblate 3.11.3

Veröffentlicht am 11. März 2020.

- Fixed searching for fields with certain priority.
- Fixed predefined query for recently added strings.
- Fixed searching returning duplicate matches.
- Fixed notifications rendering in Gmail.
- Fixed reverting changes from the history.
- Links zu Ereignissen in Zusammenfassungen-Benachrichtigungen hinzugefügt.

- Fixed email for account removal confirmation.
- Unterstützung für Slack-Authentifizierung in Docker-Containern hinzugefügt.
- Avoid sending notifications for not subscribed languages.
- Include Celery queues in performance overview.
- Die Links zu den Dokumentationen für die Erweiterungen korrigiert.
- Reduced false negatives for unchanged translation check.
- Raised bleach dependency to address CVE-2020-6802.
- Fixed listing project level changes in history.
- Die Ungültigkeit von Statistiken in einigen Ausnahmefällen behoben.
- Fixed searching for certain string states.
- Improved format string checks behavior on missing percent.
- Die Authentifizierung bei einigen Drittanbietern korrigiert.

4.48.2 Weblate 3.11.2

Veröffentlicht am 22. Februar 2020.

- Fixed rendering of suggestions.
- Fixed some strings wrongly reported as having no words.

4.48.3 Weblate 3.11.1

Veröffentlicht am 20. Februar 2020.

- Documented Celery setup changes.
- Improved filename validation on component creation.
- Fixed minimal versions of some dependencies.
- Fixed adding groups with certain Django versions.
- Fixed manual pushing to upstream repository.
- Improved glossary matching.

4.48.4 Weblate 3.11

Veröffentlicht am 17. Februar 2020.

- Allow using VCS push URL during component creation via API.
- Rendered width check now shows image with the render.
- Fixed links in notifications e-mails.
- Improved look of plaintext e-mails.
- Display ignored checks and allow to make them active again.
- Anzeige von benachbarten Schlüsseln bei einsprachigen Übersetzungen.
- Unterstützung für die Gruppierung von Zeichenketten hinzugefügt.
- Recommend upgrade to new Weblate versions in the system checks.
- Provide more detailed analysis for duplicate language alert.

- Include more detailed license info on the project pages.
- Automatically unshallow local copies if needed.
- Fixed download of strings needing action.
- New alert to warn about using the same file mask twice.
- Improve XML placeables extraction.
- The `SINGLE_PROJECT` can now enforce redirection to chosen project.
- Added option to resolve comments.
- Added bulk editing of flags.
- Unterstützung für Labels hinzugefügt.
- Erweiterung für Massenbearbeitung hinzugefügt.
- Option für *Qualitätsprüfungen erzwingen* hinzugefügt.
- Increased default validity of confirmation links.
- Improved Matomo integration.
- Fixed *Ist übersetzt worden* to correctly handle source string change.
- Extended automatic updates configuration by `AUTO_UPDATE`.
- LINGUAS-Erweiterungen synchronisieren jetzt vollständig die Übersetzungen in Weblate.

4.48.5 Weblate 3.10.3

Veröffentlicht am 18. Januar 2020.

- Unterstützung für translate-toolkit 2.5.0.

4.48.6 Weblate 3.10.2

Veröffentlicht am 18. Januar 2020.

- Add lock indication to projects.
- CSS-Fehler behoben, der in einigen Webbrowsern ein Flackern verursachte.
- Fixed searching on systems with non-English locales.
- Improved repository matching for GitHub and Bitbucket hooks.
- Fixed data migration on some Python 2.7 installations.
- Allow configuration of Git shallow cloning.
- Improved background notification processing.
- Fixed broken form submission when navigating back in web browser.
- Neue Erweiterung zur Konfiguration der YAML-Formatierung.
- Fixed same plurals check to not fire on single plural form languages.
- Fixed regex search on some fields.

4.48.7 Weblate 3.10.1

Veröffentlicht am 9. Januar 2020.

- Extended API with translation creation.
- Mehrere Ausnahmefälle bei Datenmigrationen behoben.
- Compatibility with Django 3.0.
- Verbesserte Leistung der Datenbereinigung.
- Unterstützung für anpassbare security.txt hinzugefügt.
- Improved breadcrumbs in changelog.
- Verbesserte Auflistung der Übersetzungen auf der Übersichtsseite.
- Improved HTTP responses for webhooks.
- Unterstützung für GitLab-Merge-Requests in Docker-Containern hinzugefügt.

4.48.8 Weblate 3.10

Veröffentlicht am 20. Dezember 2019.

- Improved application user interface.
- Added doublespace check.
- Fixed creating new languages.
- Avoid sending auditlog notifications to deleted e-mails.
- Unterstützung für schreibgeschützte Zeichenketten hinzugefügt.
- Unterstützung für Markdown in Kommentaren hinzugefügt.
- Allow placing translation instruction text in project info.
- Add copy to clipboard for secondary languages.
- Verbesserte Unterstützung für Mercurial.
- Improved Git repository fetching performance.
- Add search lookup for age of string.
- Anzeige der Ausgangssprache für alle Übersetzungen.
- Anzeige eines Kontexts für benachbarte Zeichenketten.
- Unterstützung für Benachrichtigungen bei Repository-Operationen hinzugefügt.
- Verbesserte Auflistung der Übersetzungen.
- Erweiterte Suchmöglichkeiten.
- Unterstützung für die automatische Übersetzung von Zeichenketten, die zur Bearbeitung markiert sind, hinzugefügt.
- Vermeidung des Versands von doppelten Benachrichtigungen für verknüpfte Komponentenwarnungen.
- Standardmeldung für Merge Request verbessert.
- Bessere Anzeige des Zeichenkettenstatus im Zen-Modus.
- Unterstützung für weitere Sprachen in Yandex Translate hinzugefügt.
- Das Aussehen der Benachrichtigungs-E-Mails verbessert.
- Eine Wahlmöglichkeit für die Übersetzungslizenz hinzugefügt.

4.48.9 Weblate 3.9.1

Veröffentlicht am 28. Oktober 2019.

- Einige nicht benötigte Dateien aus Backups entfernt.
- Fixed potential crash in reports.
- Fixed cross database migration failure.
- Unterstützung für erzwungenes Pushen von Git-Repositorys hinzugefügt.
- Reduced risk of registration token invalidation.
- Fixed account removal hitting rate limiter.
- Added search based on priority.
- Fixed possible crash on adding strings to JSON file.
- Safe HTML check and fixup now honor source string markup.
- Avoid sending notifications to invited and deleted users.
- Fix SSL connection to redis in Celery in Docker container.

4.48.10 Weblate 3.9

Veröffentlicht am 15. Oktober 2019.

- Include Weblate metadata in downloaded files.
- Improved UI for failing checks.
- Indicate missing strings in format checks.
- Separate check for French punctuation spacing.
- Add support for fixing some of quality checks errors.
- Add separate permission to create new projects.
- Extend stats for char counts.
- Improve support for Java style language codes.
- Added new generic check for placeholders.
- Added support for WebExtension JSON placeholders.
- Unterstützung für Flat XML-Format hinzugefügt.
- Extended API with project, component and translation removal and creation.
- Unterstützung für Gitea- und Gitee-Webhooks hinzugefügt.
- Added new custom regex based check.
- Allow to configure contributing to shared translation memory.
- Added ZIP download for more translation files.
- Make XLIFF standard compliant parsing of maxwidth and font.
- Added new check and fixer for safe HTML markup for translating web applications.
- Add component alert on unsupported configuration.
- Erweiterung für automatische Übersetzung zu Bootstrap-Übersetzungen hinzugefügt.
- Extend automatic translation to add suggestions.
- Erweiterungsparameter werden in der Übersicht angezeigt.

- Sentry is now supported through modern Sentry SDK instead of Raven.
- Changed example settings to be better fit for production environment.
- Automatische Backups mit BorgBackup hinzugefügt.
- Bereinigungsvorgang für die RESX-Erweiterung zur Vermeidung unerwünschter Dateiaktualisierungen aufgeteilt.
- Added advanced search capabilities.
- Allow users to download their own reports.
- Added localization guide to help configuring components.
- Unterstützung für GitLab-Merge-Requests hinzugefügt.
- Improved display of repository status.
- Perform automated translation in the background.

4.48.11 Weblate 3.8

Veröffentlicht am 15. August 2019.

- Added support for simplified creating of similar components.
- Added support for parsing translation flags from the XML based file formats.
- Log exceptions into Celery log.
- Verbesserte Leistung von Erweiterungen mit Repository-Spezifikation.
- Das Aussehen der Benachrichtigungs-E-Mails verbessert.
- Fixed password reset behavior.
- Improved performance on most of translation pages.
- Fixed listing of languages not known to Weblate.
- Unterstützung des Klonens von Erweiterungen für erkannte Komponenten hinzugefügt.
- Add support for replacing file content with uploaded.
- Add support for translating non VCS based content.
- Added OpenGraph widget image to use on social networks.
- Unterstützung für animierte Bildschirmfotos hinzugefügt.
- Improved handling of monolingual XLIFF files.
- Avoid sending multiple notifications for single event.
- Add support for filtering changes.
- Extended predefined periods for reporting.
- Added webhook support for Azure Repos.
- New opt-in notifications on pending suggestions or untranslated strings.
- Add one click unsubscribe link to notification e-mails.
- Fixed false positives with Has been translated check.
- New management interface for admins.
- String priority can now be specified using flags.
- Added language management views.
- Add checks for Qt library and Ruby format strings.

- Added configuration to better fit single project installations.
- Notify about new string on source string change on monolingual translations.
- Added separate view for translation memory with search capability.

4.48.12 Weblate 3.7.1

Veröffentlicht am 28. Juni 2019.

- Documentation updates.
- Fixed some requirements constraints.
- Updated language database.
- Localization updates.
- Various user interface tweaks.
- Improved handling of unsupported but discovered translation files.
- More verbosely report missing file format requirements.

4.48.13 Weblate 3.7

Veröffentlicht am 21. Juni 2019.

- Added separate Celery queue for notifications.
- Use consistent look with application for API browsing.
- Include approved stats in the reports.
- Report progress when updating translation component.
- Allow to abort running background component update.
- Extend template language for filename manipulations.
- Use templates for editor link and repository browser URL.
- Indicate max length and current characters count when editing translation.
- Improved handling of abbreviations in unchanged translation check.
- Refreshed landing page for new contributors.
- Unterstützung für die Konfiguration der msgmerge-Erweiterung hinzugefügt.
- Delay opening SMTP connection when sending notifications.
- Improved error logging.
- Allow custom location in MO generating add-on.
- Erweiterungen um alte Vorschläge oder Kommentare zu bereinigen hinzugefügt.
- Added option to enable horizontal mode in the Zen editor.
- Improved import performance with many linked components.
- Fixed examples installation in some cases.
- Improved rendering of alerts in changes.
- Added new horizontal stats widget.
- Improved format strings check on plurals.
- Added font management tool.

- New check for rendered text dimensions.
- Added support for subtitle formats.
- Include overall completion stats for languages.
- Added reporting at project and global scope.
- Improved user interface when showing translation status.
- New Weblate logo and color scheme.
- Neues Aussehen der Bitmap-Symbole.

4.48.14 Weblate 3.6.1

Veröffentlicht am 26. April 2019.

- Improved handling of monolingual XLIFF files.
- Die Zusammenfassungs-Benachrichtigungen in einigen Ausnahmefällen korrigiert.
- Fehlermeldung des Erweiterungsskripts behoben.
- Fixed generating MO file for monolingual PO files.
- Fixed display of uninstalled checks.
- Indicate administered projects on project listing.
- Allow update to recover from missing VCS repository.

4.48.15 Weblate 3.6

Veröffentlicht am 20. April 2019.

- Add support for downloading user data.
- Erweiterungen werden jetzt automatisch bei der Installation ausgelöst.
- Improved instructions for resolving merge conflicts.
- Die Bereinigungserweiterung ist jetzt mit der Übersetzung von App-Store-Metadaten kompatibel.
- Configurable language code syntax when adding new translations.
- Warn about using Python 2 with planned termination of support in April 2020.
- Extract special characters from the source string for visual keyboard.
- Die Übersetzerstatistik wurde erweitert, um sowohl die Anzahl der Ausgangs- als auch der Zielzeichenketten zu berücksichtigen.
- Administratoren und Konsistenz-Erweiterungen können jetzt Übersetzungen hinzufügen, auch wenn sie für Benutzer deaktiviert sind.
- Die Beschreibung der Umschaltfunktion zur Deaktivierung der `Language-Team` Header-Manipulation korrigiert.
- Notify users mentioned in comments.
- Removed file format autodetection from component setup.
- Fixed generating MO file for monolingual PO files.
- Zusammenfassungs-Benachrichtigungen hinzugefügt.
- Added support for muting component notifications.
- Added notifications for new alerts, whiteboard messages or components.

- Notifications for administered projects can now be configured.
- Improved handling of three letter language codes.

4.48.16 Weblate 3.5.1

Veröffentlicht am 10. März 2019.

- Fixed Celery systemd unit example.
- Fixed notifications from HTTP repositories with login.
- Fixed race condition in editing source string for monolingual translations.
- Die Ausgabe der fehlgeschlagenen Erweiterungsausführung wird in die Protokolle aufgenommen.
- Improved validation of choices for adding new language.
- Allow to edit file format in component settings.
- Update installation instructions to prefer Python 3.
- Performance and consistency improvements for loading translations.
- Microsoft-Terminologiedienst mit aktuellen Zeep-Versionen kompatibel gemacht.
- Localization updates.

4.48.17 Weblate 3.5

Veröffentlicht am 3. März 2019.

- Improved performance of built-in translation memory.
- Added interface to manage global translation memory.
- Verbesserte Warnung bei schlechtem Komponentenstatus.
- Added user interface to manage whiteboard messages.
- Commit-Nachricht für eine Erweiterung kann jetzt konfiguriert werden.
- Reduce number of commits when updating upstream repository.
- Fixed possible metadata loss when moving component between projects.
- Improved navigation in the Zen mode.
- Added several new quality checks (Markdown related and URL).
- Added support for app store metadata files.
- Unterstützung für das Umschalten zwischen GitHub- und Gerrit-Integration hinzugefügt.
- Added check for Kashida letters.
- Option hinzugefügt, um Commits basierend auf Autoren zusammenzufassen.
- Verbesserte Unterstützung für das XLSX-Dateiformat.
- Compatibility with Tesseract 4.0.
- Abrechnungserweiterung entfernt jetzt Projekte für unbezahlte Abrechnungen nach 45 Tagen.

4.48.18 Weblate 3.4

Veröffentlicht am 22. Januar 2019.

- Added support for XLIFF placeholders.
- Celery can now utilize multiple task queues.
- Added support for renaming and moving projects and components.
- Include characters counts in reports.
- Added guided adding of translation components with automatic detection of translation files.
- Customizable merge commit messages for Git.
- Added visual indication of component alerts in navigation.
- Improved performance of loading translation files.
- Neue Erweiterung um Commits vor dem Pushen zusammenzufassen.
- Improved displaying of translation changes.
- Changed default merge style to rebase and made that configurable.
- Better handle private use subtags in language code.
- Improved performance of fulltext index updates.
- Extended file upload API to support more parameters.

4.48.19 Weblate 3.3

Veröffentlicht am 30. November 2018.

- Unterstützung für das Entfernen von Komponenten und Projekten hinzugefügt.
- Improved performance for some monolingual translations.
- Added translation component alerts to highlight problems with a translation.
- Expose XLIFF string resname as context when available.
- Added support for XLIFF states.
- Added check for non writable files in DATA_DIR.
- Improved CSV export for changes.

4.48.20 Weblate 3.2.2

Veröffentlicht am 20. Oktober 2018.

- Remove no longer needed Babel dependency.
- Sprachdefinitionen aktualisiert.
- Verbesserung der Dokumentation für Erweiterungen, LDAP und Celery.
- Fixed enabling new dos-eol and auto-java-messageformat flags.
- Fixed running setup.py test from PyPI package.
- Improved plurals handling.
- Fehler beim Hochladen von Übersetzungen über die API in einigen Ausnahmefällen behoben.
- Fixed updating Git configuration in case it was changed manually.

4.48.21 Weblate 3.2.1

Veröffentlicht am 10. Oktober 2018.

- Document dependency on backports.csv on Python 2.7.
- Fix running tests under root.
- Improved error handling in gitexport module.
- Fixed progress reporting for newly added languages.
- Correctly report Celery worker errors to Sentry.
- Fixed creating new translations with Qt Linguist.
- Fixed occasional fulltext index update failures.
- Improved validation when creating new components.
- Unterstützung für die Bereinigung alter Übersetzungsvorschläge hinzugefügt.

4.48.22 Weblate 3.2

Veröffentlicht am 6. Oktober 2018.

- Hinzufügen des Verwaltungsbefehls `install_addon` für die automatische Installation von Erweiterungen.
- Allow more fine grained ratelimit settings.
- Unterstützung für den Export und Import von Excel-Dateien hinzugefügt.
- Improve component cleanup in case of multiple component discovery add-ons.
- Rewritten Microsoft Terminology machine translation backend.
- Weblate now uses Celery to offload some processing.
- Improved search capabilities and added regular expression search.
- Added support for Youdao Zhiyun API machine translation.
- Added support for Baidu API machine translation.
- Integrated maintenance and cleanup tasks using Celery.
- Improved performance of loading translations by almost 25%.
- Removed support for merging headers on upload.
- Removed support for custom commit messages.
- Configurable editing mode (zen/full).
- Unterstützung für Fehlerberichterstattung an Sentry hinzugefügt.
- Added support for automated daily update of repositories.
- Added support for creating projects and components by users.
- Der integrierte Übersetzungsspeicher speichert jetzt automatisch durchgeführte Übersetzungen.
- Users and projects can import their existing translation memories.
- Bessere Verwaltung von zusammenhängenden Zeichenketten für Bildschirmfotos.
- Added support for checking Java MessageFormat.

See [3.2 milestone on GitHub](#) for detailed list of addressed issues.

4.48.23 Weblate 3.1.1

Veröffentlicht am 27. Juli 2018.

- Fix testsuite failure on some setups.

4.48.24 Weblate 3.1

Veröffentlicht am 27. Juli 2018.

- Upgrades von älteren Versionen als 3.0.1 werden nicht unterstützt.
- Allow to override default commit messages from settings.
- Improve webhooks compatibility with self hosted environments.
- Unterstützung für Amazon Translate hinzugefügt.
- Compatibility with Django 2.1.
- Django system checks are now used to diagnose problems with installation.
- Removed support for soon shutdown libavatar service.
- Neue Erweiterung, um unveränderte Übersetzungen als bearbeitungsbedürftig zu markieren.
- Add support for jumping to specific location while translating.
- Downloaded translations can now be customized.
- Improved calculation of string similarity in translation memory matches.
- Added support by signing Git commits by GnuPG.

4.48.25 Weblate 3.0.1

Veröffentlicht am 10. Juni 2018.

- Fixed possible migration issue from 2.20.
- Localization updates.
- Removed obsolete hook examples.
- Improved caching documentation.
- Fixed displaying of admin documentation.
- Improved handling of long language names.

4.48.26 Weblate 3.0

Veröffentlicht am 1. Juni 2018.

- Rewritten access control.
- Several code cleanups that lead to moved and renamed modules.
- New add-on for automatic component discovery.
- The import_project management command has now slightly different parameters.
- Grundlegende Unterstützung für Windows RC-Dateien hinzugefügt.
- New add-on to store contributor names in PO file headers.
- The per component hook scripts are removed, use add-ons instead.
- Add support for collecting contributor agreements.

- Access control changes are now tracked in history.
- Neue Erweiterung, um sicherzustellen, dass alle Komponenten in einem Projekt die gleichen Übersetzungen haben.
- Support for more variables in commit message templates.
- Add support for providing additional textual context.

4.49 Weblate 2.x-Reihe

4.49.1 Weblate 2.20

Veröffentlicht am 4. April 2018.

- Improved speed of cloning subversion repositories.
- Changed repository locking to use third party library.
- Added support for downloading only strings needing action.
- Added support for searching in several languages at once.
- New add-on to configure gettext output wrapping.
- Neue Erweiterung zur Konfiguration der JSON-Formatierung.
- Unterstützung für die Authentifizierung in API mit RFC 6750 kompatibler Bearer-Authentifizierung hinzugefügt.
- Unterstützung für die automatische Übersetzung durch maschinelle Übersetzungsdienste hinzugefügt.
- Unterstützung für HTML-Markup in Whiteboard-Nachrichten hinzugefügt.
- Unterstützung für die Massенbearbeitung des Zeichenkettenstatus hinzugefügt.
- Translate-toolkit at least 2.3.0 is now required, older versions are no longer supported.
- Integrierter Übersetzungsspeicher hinzugefügt.
- Added componentlists overview to dashboard and per component list overview pages.
- Unterstützung für den maschinellen Übersetzungsdienst DeepL hinzugefügt.
- Die Ergebnisse der maschinellen Übersetzung werden jetzt in Weblate zwischengespeichert.
- Unterstützung für die Neuordnung von bestätigten Änderungen hinzugefügt.

4.49.2 Weblate 2.19.1

Veröffentlicht am 20. Februar 2018.

- Migrationsproblem beim Upgrade von 2.18 behoben.
- Verbesserte Validierung der Dateiupload-API.

4.49.3 Weblate 2.19

Veröffentlicht am 15. Februar 2018.

- Fixed imports across some file formats.
- Display human friendly browser information in audit log.
- Added TMX exporter for files.
- Various performance improvements for loading translation files.
- Added option to disable access management in Weblate in favor of Django one.
- Improved glossary lookup speed for large strings.
- Compatibility with django_auth_ldap 1.3.0.
- Configuration errors are now stored and reported persistently.
- Honor ignore flags in whitespace autofixer.
- Improved compatibility with some Subversion setups.
- Verbesserte integrierte maschinelle Übersetzungsdienste.
- Unterstützung für den SAP Translation Hub Dienst hinzugefügt.
- Unterstützung für den Microsoft-Terminologiedienst hinzugefügt.
- Unterstützung für Werbung in Benachrichtigungs-E-Mails entfernt.
- Improved translation progress reporting at language level.
- Verbesserte Unterstützung für verschiedene Pluralformeln.
- Added support for Subversion repositories not using stdlayout.
- Erweiterungen zum Anpassen von Übersetzungsabläufen hinzugefügt.

4.49.4 Weblate 2.18

Veröffentlicht am 15. Dezember 2017.

- Erweiterte Übersetzerstatistik.
- Improved configuration of special characters virtual keyboard.
- Unterstützung für das DTD-Dateiformat hinzugefügt.
- Die Tastaturkürzel wurden geändert, um nicht mehr so leicht mit denen des Browsers/Systems zu kollidieren.
- Improved support for approved flag in XLIFF files.
- Added support for not wrapping long strings in gettext PO files.
- Schaltfläche zum Kopieren des Permalinks für die aktuelle Übersetzung hinzugefügt.
- Die Unterstützung für Django 1.10 eingestellt und die Unterstützung für Django 2.0 hinzugefügt.
- Removed locking of translations while translating.
- Added support for adding new strings to monolingual translations.
- Added support for translation workflows with dedicated reviewers.

4.49.5 Weblate 2.17.1

Veröffentlicht am 13. Oktober 2017.

- Fixed running testsuite in some specific situations.
- Locales updates.

4.49.6 Weblate 2.17

Veröffentlicht am 13. Oktober 2017.

- Weblate by default does shallow Git clones now.
- Improved performance when updating large translation files.
- Added support for blocking certain e-mails from registration.
- Users can now delete their own comments.
- Added preview step to search and replace feature.
- Client side persistence of settings in search and upload forms.
- Erweiterte Suchmöglichkeiten.
- More fine grained per project ACL configuration.
- Default value of BASE_DIR has been changed.
- Added two step account removal to prevent accidental removal.
- Project access control settings is now editable.
- Added optional spam protection for suggestions using Akismet.

4.49.7 Weblate 2.16

Veröffentlicht am 11. August 2017.

- Various performance improvements.
- Added support for nested JSON format.
- Added support for WebExtension JSON format.
- Die Authentifizierung des Git-Exporters korrigiert.
- Improved CSV import in certain situations.
- Improved look of Other translations widget.
- The max-length checks is now enforcing length of text in form.
- Make the commit_pending age configurable per component.
- Various user interface cleanups.
- Die Komponenten-, Projekt- und plattformweite Suche nach Übersetzungen korrigiert.

4.49.8 Weblate 2.15

Veröffentlicht am 30. Juni 2017.

- Show more related translations in other translations.
- Add option to see translations of current string to other languages.
- Use 4 plural forms for Lithuanian by default.
- Fixed upload for monolingual files of different format.
- Verbesserte Fehlermeldungen bei fehlgeschlagener Authentifizierung.
- Keep page state when removing word from glossary.
- Added direct link to edit secondary language translation.
- Added Perl format quality check.
- Added support for rejecting reused passwords.
- Erweiterte Symbolleiste für die Bearbeitung von Sprachen mit linksläufiger Schreibrichtung.

4.49.9 Weblate 2.14.1

Veröffentlicht am 24. Mai 2017.

- Fixed possible error when paginating search results.
- Die Migration von älteren Versionen in einigen Ausnahmefällen korrigiert.
- Fixed possible CSRF on project watch and unwatch.
- The password reset no longer authenticates user.
- Fixed possible CAPTCHA bypass on forgotten password.

4.49.10 Weblate 2.14

Veröffentlicht am 17. Mai 2017.

- Add glossary entries using AJAX.
- The logout now uses POST to avoid CSRF.
- The API key token reset now uses POST to avoid CSRF.
- Weblate sets Content-Security-Policy by default.
- The local editor URL is validated to avoid self-XSS.
- The password is now validated against common flaws by default.
- Notify users about important activity with their account such as password change.
- The CSV exports now escape potential formulas.
- Various minor improvements in security.
- Die Authentifizierungsversuche sind nun ratenbegrenzt.
- Suggestion content is stored in the history.
- Store important account activity in audit log.
- Ask for password confirmation when removing account or adding new associations.
- Show time when suggestion has been made.
- There is new quality check for trailing semicolon.

- Ensure that search links can be shared.
- Aufnahme von Informationen zur Ausgangszeichenkette und Bildschirmfotos in die API.
- Allow to overwrite translations through API upload.

4.49.11 Weblate 2.13.1

Veröffentlicht am 12. April 2017.

- Fixed listing of managed projects in profile.
- Fixed migration issue where some permissions were missing.
- Fixed listing of current file format in translation download.
- HTTP 404 zurückgeben, wenn versucht wird, auf ein Projekt zuzugreifen, für das der Benutzer keine Berechtigungen hat.

4.49.12 Weblate 2.13

Veröffentlicht am 12. April 2017.

- Fixed quality checks on translation templates.
- Added quality check to trigger on losing translation.
- Add option to view pending suggestions from user.
- Add option to automatically build component lists.
- Default dashboard for unauthenticated users can be configured.
- Add option to browse 25 random strings for review.
- History now indicates string change.
- Bessere Fehlerberichterstattung beim Hinzufügen einer neuen Übersetzung.
- Added per language search within project.
- Group ACLs can now be limited to certain permissions.
- The per project ACLs are now implemented using Group ACL.
- Eine feinere Kontrolle der Berechtigungen hinzugefügt.
- Various minor UI improvements.

4.49.13 Weblate 2.12

Veröffentlicht am 3. März 2017.

- Verbesserte Adminoberfläche für Gruppen.
- Added support for Yandex Translate API.
- Verbesserte Geschwindigkeit der plattformweiten Suche.
- Added project and component wide search.
- Added project and component wide search and replace.
- Improved rendering of inconsistent translations.
- Added support for opening source files in local editor.
- Added support for configuring visual keyboard with special characters.

- Verbesserte Bildschirmfoto-Verwaltung mit OCR-Unterstützung für übereinstimmende Ausgangszeichenketten.
- Default commit message now includes translation information and URL.
- Unterstützung für Joomla-Übersetzungsformat hinzugefügt.
- Improved reliability of import across file formats.

4.49.14 Weblate 2.11

Veröffentlicht am 31. Januar 2017.

- Include language detailed information on language page.
- Mercurial backend improvements.
- Added option to specify translation component priority.
- More consistent usage of Group ACL even with less used permissions.
- Added WL_BRANCH variable to hook scripts.
- Improved developer documentation.
- Bessere Kompatibilität mit verschiedenen Git-Versionen in der Git-Export-Erweiterung.
- Included per project and component stats.
- Added language code mapping for better support of Microsoft Translate API.
- Moved fulltext cleanup to background job to make translation removal faster.
- Fixed displaying of plural source for languages with single plural form.
- Improved error handling in import_project.
- Various performance improvements.

4.49.15 Weblate 2.10.1

Veröffentlicht am 20. Januar 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

4.49.16 Weblate 2.10

Veröffentlicht am 15. Dezember 2016.

- Added quality check to check whether plurals are translated differently.
- GitHub-Hooks für Repositories mit Authentifizierung korrigiert.
- Added optional Git exporter module.
- Support for Microsoft Cognitive Services Translator API.
- Simplified project and component user interface.
- Added automatic fix to remove control characters.
- Added per language overview to project.
- Unterstützung für CSV-Export hinzugefügt.
- Added CSV download for stats.
- Matrixansicht für schnellen Überblick über alle Übersetzungen hinzugefügt.

- Added basic API for changes and strings.
- Added support for Apertium APy server for machine translations.

4.49.17 Weblate 2.9

Veröffentlicht am 4. November 2016.

- Extended parameters for createadmin management command.
- Extended import_json to be able to handle with existing components.
- Added support for YAML files.
- Project owners can now configure translation component and project details.
- Use „Watched“ instead of „Subscribed“ projects.
- Projects can be watched directly from project page.
- Added multi language status widget.
- Highlight secondary language if not showing source.
- Record suggestion deletion in history.
- Improved UX of languages selection in profile.
- Fixed showing whiteboard messages for component.
- Keep preferences tab selected after saving.
- Show source string comment more prominently.
- Automatically install Gettext PO merge driver for Git repositories.
- Added search and replace feature.
- Unterstützung für das Hochladen von visuellem Kontext (Bildschirmfotos) für Übersetzungen hinzugefügt.

4.49.18 Weblate 2.8

Veröffentlicht am 31. August 2016.

- Verbesserung der Dokumentation.
- Translations.
- Updated bundled JavaScript libraries.
- Added list_translators management command.
- Django 1.8 is no longer supported.
- Fixed compatibility with Django 1.10.
- Added Subversion support.
- Separated XML validity check from XML mismatched tags.
- Fixed API to honor HIDE_REPO_CREDENTIALS settings.
- Show source change in Zen mode.
- Alt+PageUp/PageDown/Home/End now works in Zen mode as well.
- Add tooltip showing exact time of changes.
- Add option to select filters and search from translation page.
- Added UI for translation removal.

- Improved behavior when inserting placeables.
- Fixed auto locking issues in Zen mode.

4.49.19 Weblate 2.7

Veröffentlicht am 10. Juli 2016.

- Removed Google web translate machine translation.
- Improved commit message when adding translation.
- Fixed Google Translate API for Hebrew language.
- Compatibility with Mercurial 3.8.
- Added import_json management command.
- Correct ordering of listed translations.
- Show full suggestion text, not only a diff.
- Erweiterung der API (detaillierter Repository-Status, Statistiken, ...).
- Testsuite no longer requires network access to test repositories.

4.49.20 Weblate 2.6

Veröffentlicht am 28. April 2016.

- Fixed validation of components with language filter.
- Verbesserte Unterstützung für XLIFF-Dateien.
- Fixed machine translation for non English sources.
- Added REST API.
- Django 1.10 compatibility.
- Added categories to whiteboard messages.

4.49.21 Weblate 2.5

Veröffentlicht am 10. März 2016.

- Fixed automatic translation for project owners.
- Improved performance of commit and push operations.
- New management command to add suggestions from command-line.
- Unterstützung für das Zusammenführen von Kommentaren beim Hochladen von Dateien hinzugefügt.
- Unterstützung für einige GNU-Erweiterungen des C-printf-Formats hinzugefügt.
- Verbesserung der Dokumentation.
- Unterstützung für die Erstellung von Übersetzungsnachweisen hinzugefügt.
- Unterstützung für die Erstellung von Übersetzerstatistiken hinzugefügt.
- Die plattformweite Suche kann nur in einer Sprache suchen.
- Improve quality checks for Armenian.
- Support for starting translation components without existing translations.
- Support for adding new translations in Qt TS.

- Improved support for translating PHP files.
- Performance improvements for quality checks.
- Die plattformweite Suche nach fehlgeschlagenen Qualitätsprüfungen korrigiert.
- Added option to specify source language.
- Verbesserte Unterstützung für XLIFF-Dateien.
- Extended list of options for import_project.
- Improved targeting for whiteboard messages.
- Support for automatic translation across projects.
- Optimized fulltext search index.
- Added management command for auto translation.
- Added placeables highlighting.
- Added keyboard shortcuts for placeables, checks and machine translations.
- Improved translation locking.
- Added quality check for AngularJS interpolation.
- Added extensive group based ACLs.
- Clarified terminology on strings needing edit (formerly fuzzy).
- Clarified terminology on strings needing action and untranslated strings.
- Unterstützung für Python 3.
- Die Unterstützung für Django 1.7 eingestellt.
- Dropped dependency on msginit for creating new gettext PO files.
- Added configurable dashboard views.
- Improved notifications on parse errors.
- Added option to import components with duplicate name to import_project.
- Improved support for translating PHP files.
- Added XLIFF export for dictionary.
- Added XLIFF and gettext PO export for all translations.
- Verbesserung der Dokumentation.
- Unterstützung für konfigurierbare automatische Gruppenzuweisungen hinzugefügt.
- Improved adding of new translations.

4.49.22 Weblate 2.4

Veröffentlicht am 20. September 2015.

- Verbesserte Unterstützung für PHP-Dateien.
- Ability to add ACL to anonymous user.
- Improved configurability of import_project command.
- Added CSV dump of history.
- Avoid copy/paste errors with whitespace characters.
- Unterstützung für Bitbucket-Webhooks hinzugefügt.
- Tighter control on fuzzy strings on translation upload.

- Several URLs have changed, you might have to update your bookmarks.
- Hook scripts are executed with VCS root as current directory.
- Hook scripts are executed with environment variables describing current component.
- Add management command to optimize fulltext index.
- Unterstützung für Fehlerberichterstattung in Rollbar hinzugefügt.
- Projects now can have multiple owners.
- Project owners can manage themselves.
- Unterstützung für das in gettext PO verwendete `javascript-format` hinzugefügt.
- Support for adding new translations in XLIFF.
- Improved file format autodetection.
- Extended keyboard shortcuts.
- Improved dictionary matching for several languages.
- Improved layout of most of pages.
- Support for adding words to dictionary while translating.
- Added support for filtering languages to be managed by Weblate.
- Added support for translating and importing CSV files.
- Rewritten handling of static files.
- Direct login/registration links to third-party service if that's the only one.
- Commit pending changes on account removal.
- Add management command to change site name.
- Add option to configure default committer.
- Add hook after adding new translation.
- Add option to specify multiple files to add to commit.

4.49.23 Weblate 2.3

Veröffentlicht am 22. Mai 2015.

- Die Unterstützung für Django 1.6 und South-Migrationen eingestellt.
- Support for adding new translations when using Java Property files.
- Allow to accept suggestion without editing.
- Verbesserte Unterstützung für Google OAuth 2.0.
- Unterstützung für Microsoft .resx-Dateien hinzugefügt.
- Tuned default robots.txt to disallow big crawling of translations.
- Simplified workflow for accepting suggestions.
- Added project owners who always receive important notifications.
- Allow to disable editing of monolingual template.
- More detailed repository status view.
- Direct link for editing template when changing translation.
- Allow to add more permissions to project owners.
- Allow to show secondary language in Zen mode.

- Support for hiding source string in favor of secondary language.

4.49.24 Weblate 2.2

Veröffentlicht am 19. Februar 2015.

- Leistungsverbesserungen.
- Fulltext search on location and comments fields.
- New SVG/JavaScript-based activity charts.
- Support for Django 1.8.
- Support for deleting comments.
- Eigenes SVG-Symbol hinzugefügt.
- Unterstützung für Google Analytics hinzugefügt.
- Improved handling of translation filenames.
- Unterstützung für einsprachige JSON-Übersetzungen hinzugefügt.
- Record component locking in a history.
- Support for editing source (template) language for monolingual translations.
- Grundlegende Unterstützung für Gerrit hinzugefügt.

4.49.25 Weblate 2.1

Veröffentlicht am 5. Dezember 2014.

- Unterstützung für Mercurial-Repositorys hinzugefügt.
- Replaced Glyphicon font by Awesome.
- Icons für Authentifizierungsdienste der sozialen Medien hinzugefügt.
- Better consistency of button colors and icons.
- Verbesserung der Dokumentation.
- Verschiedene Fehlerbehebungen.
- Automatic hiding of columns in translation listing for small screens.
- Changed configuration of filesystem paths.
- Improved SSH keys handling and storage.
- Improved repository locking.
- Customizable quality checks per source string.
- Allow to hide completed translations from dashboard.

4.49.26 Weblate 2.0

Veröffentlicht am 6. November 2014.

- New responsive UI using Bootstrap.
- Rewritten VCS backend.
- Verbesserung der Dokumentation.
- Whiteboard für plattformweite Meldungen hinzugefügt.
- Configurable strings priority.
- Unterstützung für das JSON-Dateiformat hinzugefügt.
- Fixed generating mo files in certain cases.
- Unterstützung für GitLab-Benachrichtigungen hinzugefügt.
- Added support for disabling translation suggestions.
- Django 1.7 support.
- ACL-Projekte haben jetzt eine Benutzerverwaltung.
- Extended search possibilities.
- Give more hints to translators about plurals.
- Fixed Git repository locking.
- Compatibility with older Git versions.
- Verbesserte ACL-Unterstützung.
- Added buttons for per language quotes and other special characters.
- Support for exporting stats as JSONP.

4.50 Weblate 1.x-Reihe

4.50.1 Weblate 1.9

Veröffentlicht am 6. Mai 2014.

- Django 1.6 compatibility.
- No longer maintained compatibility with Django 1.4.
- Management commands for locking/unlocking translations.
- Improved support for Qt TS files.
- Users can now delete their account.
- Avatars can be disabled.
- Merged first and last name attributes.
- Avatars are now fetched and cached server side.
- Unterstützung für Shields.io-Symbol hinzugefügt.

4.50.2 Weblate 1.8

Veröffentlicht am 7. November 2013.

- Please check manual for upgrade instructions.
- Nicer listing of project summary.
- Better visible options for sharing.
- Mehr Kontrolle über anonyme Benutzerberechtigungen.
- Supports login using third party services, check manual for more details.
- Users can login by e-mail instead of username.
- Verbesserung der Dokumentation.
- Improved source strings review.
- Searching across all strings.
- Better tracking of source strings.
- Captcha protection for registration.

4.50.3 Weblate 1.7

Veröffentlicht am 7. Oktober 2013.

- Please check manual for upgrade instructions.
- Support for checking Python brace format string.
- Per component customization of quality checks.
- Detailed per translation stats.
- Changed way of linking suggestions, checks and comments to strings.
- Users can now add text to commit message.
- Support for subscribing on new language requests.
- Support for adding new translations.
- Widgets and charts are now rendered using Pillow instead of Pango + Cairo.
- Statussymbol-Widget hinzugefügt.
- Dropped invalid text direction check.
- Changes in dictionary are now logged in history.
- Leistungsverbesserungen für die Übersetzungsansicht.

4.50.4 Weblate 1.6

Veröffentlicht am 25. Juli 2013.

- Nicer error handling on registration.
- Browsing of changes.
- Fixed sorting of machine translation suggestions.
- Improved support for MyMemory machine translation.
- Added support for Amagama machine translation.
- Various optimizations on frequently used pages.

- Highlights searched phrase in search results.
- Support for automatic fixups while saving the message.
- Tracking of translation history and option to revert it.
- Unterstützung für Google Translate API hinzugefügt.
- Unterstützung für die Verwaltung von SSH-Hostschlüsseln hinzugefügt.
- Various form validation improvements.
- Various quality checks improvements.
- Performance improvements for import.
- Added support for voting on suggestions.
- Cleanup of admin interface.

4.50.5 Weblate 1.5

Veröffentlicht am 16. April 2013.

- Please check manual for upgrade instructions.
- Added public user pages.
- Better naming of plural forms.
- Added support for TBX export of glossary.
- Unterstützung für Bitbucket-Benachrichtigungen hinzugefügt.
- Activity charts are now available for each translation, language or user.
- Extended options of `import_project` admin command.
- Compatible with Django 1.5.
- Avatars are now shown using libavatar.
- Added possibility to pretty print JSON export.
- Various performance improvements.
- Indicate failing checks or fuzzy strings in progress bars for projects or languages as well.
- Added support for custom pre-commit hooks and committing additional files.
- Rewritten search for better performance and user experience.
- Neue Oberfläche für maschinelle Übersetzungen.
- Unterstützung für einsprachige po-Dateien hinzugefügt.
- Extend amount of cached metadata to improve speed of various searches.
- Now shows word counts as well.

4.50.6 Weblate 1.4

Veröffentlicht am 23. Januar 2013.

- Fixed deleting of checks/comments on string deletion.
- Added option to disable automatic propagation of translations.
- Added option to subscribe for merge failures.
- Correctly import on projects which needs custom ttkit loader.
- Added sitemaps to allow easier access by crawlers.
- Provide direct links to string in notification e-mails or feeds.
- Various improvements to admin interface.
- Provide hints for production setup in admin interface.
- Added per language widgets and engage page.
- Improved translation locking handling.
- Show code snippets for widgets in more variants.
- Indicate failing checks or fuzzy strings in progress bars.
- More options for formatting commit message.
- Die Fehlerbehandlung bei maschinellen Übersetzungsdiensten korrigiert.
- Verbessertes Sperrverhalten der automatischen Übersetzung.
- Support for showing changes from previous source string.
- Added support for substring search.
- Various quality checks improvements.
- Support for per project ACL.
- Basic code coverage by unit tests.

4.50.7 Weblate 1.3

Veröffentlicht am 16. November 2012.

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Improved quality checks processing.
- Neue Prüfungen hinzugefügt (BB-Code, XML-Markup und Zeilenumbrüche).
- Support for optional rebasing instead of merge.
- Possibility to relocate Weblate (for example to run it under /weblate path).
- Support for manually choosing file type in case autodetection fails.
- Better support for Android resources.
- Support for generating SSH key from web interface.
- More visible data exports.
- New buttons to enter some special characters.
- Support for exporting dictionary.
- Support for locking down whole Weblate installation.

- Checks for source strings and support for source strings review.
- Support for user comments for both translations and source strings.
- Better changes log tracking.
- Changes can now be monitored using RSS.
- Verbesserte Unterstützung für Sprachen mit linksläufiger Schreibrichtung.

4.50.8 Weblate 1.2

Veröffentlicht am 14. August 2012.

- Weblate now uses South for database migration, please check upgrade instructions if you are upgrading.
- Fixed minor issues with linked git repos.
- New introduction page for engaging people with translating using Weblate.
- Widgets hinzugefügt, die zur Förderung von Übersetzungsprojekten verwendet werden können.
- Added option to reset repository to origin (for privileged users).
- Project or component can now be locked for translations.
- Possibility to disable some translations.
- Configurable options for adding new translations.
- Configuration of git commits per project.
- Simple antispam protection.
- Better layout of main page.
- Support for automatically pushing changes on every commit.
- Support for e-mail notifications of translators.
- Nur verwendete Sprachen in den Einstellungen anzeigen.
- Improved handling of not known languages when importing project.
- Support for locking translation by translator.
- Optionally maintain `Language-Team` header in po file.
- Include some statistics in about page.
- Supports (and requires) django-registration 0.8.
- Zwischenspeichern der Anzahl von Zeichenketten mit fehlgeschlagenen Qualitätsprüfungen.
- Checking of requirements during setup.
- Verbesserung der Dokumentation.

4.50.9 Weblate 1.1

Veröffentlicht am 4. Juli 2012.

- Improved several translations.
- Better validation while creating component.
- Added support for shared git repositories across components.
- Do not necessary commit on every attempt to pull remote repo.
- Added support for offloading indexing.

4.50.10 Weblate 1.0

Veröffentlicht am 10. Mai 2012.

- Improved validation while adding/saving component.
- Experimental support for Android component files (needs patched ttkit).
- Updates from hooks are run in background.
- Improved installation instructions.
- Improved navigation in dictionary.

4.51 Weblate 0.x-Reihe

4.51.1 Weblate 0.9

Veröffentlicht am 18. April 2012.

- Fixed import of unknown languages.
- Improved listing of nearby messages.
- Improved several checks.
- Documentation updates.
- Added definition for several more languages.
- Various code cleanups.
- Verbesserung der Dokumentation.
- Changed file layout.
- Update helper scripts to Django 1.4.
- Improved navigation while translating.
- Better handling of po file renames.
- Better validation while creating component.
- Integrated full setup into syncdb.
- Added list of recent changes to all translation pages.
- Check for untranslated strings ignores format string only messages.

4.51.2 Weblate 0.8

Veröffentlicht am 3. April 2012.

- Replaced own full text search with Whoosh.
- Various fixes and improvements to checks.
- New command updatechecks.
- Lot of translation updates.
- Added dictionary for storing most frequently used terms.
- Added /admin/report/ for overview of repositories status.
- Maschinelle Übersetzungsdienste blockieren das Laden von Seiten nicht mehr.
- Management interface now contains also useful actions to update data.

- Records log of changes made by users.
- Ability to postpone commit to Git to generate less commits from single user.
- Possibility to browse failing checks.
- Automatic translation using already translated strings.
- New about page showing used versions.
- Django 1.4 compatibility.
- Ability to push changes to remote repo from web interface.
- Added review of translations done by others.

4.51.3 Weblate 0.7

Veröffentlicht am 16. Februar 2012.

- Direct support for GitHub notifications.
- Added support for cleaning up orphaned checks and translations.
- Displays nearby strings while translating.
- Displays similar strings while translating.
- Improved searching for string.

4.51.4 Weblate 0.6

Veröffentlicht am 14. Februar 2012.

- Added various checks for translated messages.
- Tunable access control.
- Improved handling of translations with new lines.
- Added client side sorting of tables.
- Please check upgrading instructions in case you are upgrading.

4.51.5 Weblate 0.5

Veröffentlicht am 12. Februar 2012.

- **Unterstützung für die maschineller Übersetzung durch folgende Online-Dienste:**
 - Apertium
 - Microsoft Translator
 - MyMemory
- Several new translations.
- Improved merging of upstream changes.
- Better handle concurrent git pull and translation.
- Propagating works for fuzzy changes as well.
- Propagating works also for file upload.
- Fixed file downloads while using FastCGI (and possibly others).

4.51.6 Weblate 0.4

Veröffentlicht am 8. Februar 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

4.51.7 Weblate 0.3

Veröffentlicht am 8. Februar 2012.

- Better display of source for plural translations.
- New documentation in Sphinx format.
- Displays secondary languages while translating.
- Improved error page to give list of existing projects.
- New per language stats.

4.51.8 Weblate 0.2

Veröffentlicht am 7. Februar 2012.

- Improved validation of several forms.
- Warn users on profile upgrade.
- Remember URL for login.
- Naming of text areas while entering plural forms.
- Automatic expanding of translation area.

4.51.9 Weblate 0.1

Veröffentlicht am 6. Februar 2012.

- Erstveröffentlichung.

W

wlc, [161](#)

wlc.config, [162](#)

wlc.main, [162](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component:slug)/, 136
ANY /, 107	GET /api/components/(string:project)/(string:component:slug)/, 131
/api	GET /api/components/(string:project)/(string:component:slug)/, 133
GET /api/, 109	GET /api/components/(string:project)/(string:component:slug)/, 134
/api/addons	GET /api/components/(string:project)/(string:component:slug)/, 132
GET /api/addons/, 148	GET /api/components/(string:project)/(string:component:slug)/, 131
GET /api/addons/(int:id)/, 148	GET /api/components/(string:project)/(string:component:slug)/, 135
PUT /api/addons/(int:id)/, 149	GET /api/components/(string:project)/(string:component:slug)/, 134
DELETE /api/addons/(int:id)/, 149	POST /api/components/(string:project)/(string:component:slug)/, 148
PATCH /api/addons/(int:id)/, 148	POST /api/components/(string:project)/(string:component:slug)/, 136
/api/changes	POST /api/components/(string:project)/(string:component:slug)/, 132
GET /api/changes/, 145	POST /api/components/(string:project)/(string:component:slug)/, 133
GET /api/changes/(int:id)/, 145	POST /api/components/(string:project)/(string:component:slug)/, 134
/api/component-lists	PUT /api/components/(string:project)/(string:component:slug)/, 130
GET /api/component-lists/, 149	DELETE /api/components/(string:project)/(string:component:slug)/, 130
GET /api/component-lists/(str:slug)/, 149	DELETE /api/components/(string:project)/(string:component:slug)/, 136
POST /api/component-lists/(str:slug)/components/, 150	PATCH /api/components/(string:project)/(string:component:slug)/, 129
PUT /api/component-lists/(str:slug)/, 149	/api/groups
DELETE /api/component-lists/(str:slug)/, 150	GET /api/groups/, 114
DELETE /api/component-lists/(str:slug)/components/(string:component:slug)/, 150	GET /api/groups/(int:id)/, 114
PATCH /api/component-lists/(str:slug)/, 149	POST /api/groups/, 114
/api/components	POST /api/groups/(int:id)/changes/, 116
GET /api/components/, 126	POST /api/groups/(int:id)/components/, 115
GET /api/components/(string:project)/(string:component:slug)/, 126	
GET /api/components/(string:project)/(string:component:slug)/, 131	
GET /api/components/(string:project)/(string:component:slug)/, 131	

POST /api/groups/(int:id)/languages/, 116
 POST /api/groups/(int:id)/projects/, 116
 POST /api/groups/(int:id)/roles/, 115
 PUT /api/groups/(int:id)/, 115
 DELETE /api/groups/(int:id)/, 115
 DELETE /api/groups/(int:id)/componentlist/, 116
 DELETE /api/groups/(int:id)/components/(int:component_id)/, 116
 DELETE /api/groups/(int:id)/languages/(string:language_code_id)/, 116
 DELETE /api/groups/(int:id)/projects/(int:project_id)/, 116
 PATCH /api/groups/(int:id)/, 115

/api/languages

GET /api/languages/, 118
 GET /api/languages/(string:language)/, 118
 GET /api/languages/(string:language)/statistics/, 119
 POST /api/languages/, 118
 PUT /api/languages/(string:language)/, 119
 DELETE /api/languages/(string:language)/, 119
 PATCH /api/languages/(string:language)/, 119

/api/memory

GET /api/memory/, 142
 DELETE /api/memory/(int:memory_object_id)/, 142

/api/metrics

GET /api/metrics/, 151

/api/projects

GET /api/projects/, 120
 GET /api/projects/(string:project)/, 120
 GET /api/projects/(string:project)/changes/, 121
 GET /api/projects/(string:project)/components/, 123
 GET /api/projects/(string:project)/languages/, 125
 GET /api/projects/(string:project)/repositories/, 122
 GET /api/projects/(string:project)/statistics/, 126
 POST /api/projects/, 120
 POST /api/projects/(string:project)/components/, 123
 POST /api/projects/(string:project)/repositories/, 122

/api/roles

GET /api/roles/, 117
 POST /api/roles/, 117
 DELETE /api/roles/(int:id)/, 118
 PATCH /api/roles/(int:id)/, 117

/api/screenshots

GET /api/screenshots/, 146
 GET /api/screenshots/(int:id)/, 146
 GET /api/screenshots/(int:id)/file/, 146
 POST /api/screenshots/, 147
 POST /api/screenshots/(int:id)/file/, 146
 POST /api/screenshots/(int:id)/units/, 146
 PUT /api/screenshots/(int:id)/, 147
 DELETE /api/screenshots/(int:id)/, 148
 DELETE /api/screenshots/(int:id)/units/(int:unit_id)/, 147
 PATCH /api/screenshots/(int:id)/, 147

/api/tasks

GET /api/tasks/, 150
 GET /api/tasks/(str:uuid)/, 150

/api/translations

GET /api/translations/, 136
 GET /api/translations/(string:project)/(string:component)/, 136
 GET /api/translations/(string:project)/(string:component)/changes/, 139
 GET /api/translations/(string:project)/(string:component)/components/, 140
 GET /api/translations/(string:project)/(string:component)/languages/, 141
 GET /api/translations/(string:project)/(string:component)/repositories/, 142
 GET /api/translations/(string:project)/(string:component)/statistics/, 142
 GET /api/translations/(string:project)/(string:component)/units/, 139
 POST /api/translations/(string:project)/(string:component)/, 140
 POST /api/translations/(string:project)/(string:component)/components/, 141
 POST /api/translations/(string:project)/(string:component)/repositories/, 139

DELETE /api/translations/(string:project)/(string:component)/(string:language)/, 139
 POST /hooks/complate/, 152
 POST /hooks/pagure/, 152

/api/units

GET /api/units/, 143
 GET /api/units/(int:id)/, 143
 PUT /api/units/(int:id)/, 144
 DELETE /api/units/(int:id)/, 144
 PATCH /api/units/(int:id)/, 144

/api/users

GET /api/users/, 110
 GET /api/users/(str:username)/, 110
 GET /api/users/(str:username)/notifications/, 112
 GET /api/users/(str:username)/notifications/(int:subscription_id)/, 113
 GET /api/users/(str:username)/statistics/, 112
 POST /api/users/, 110
 POST /api/users/(str:username)/groups/, 112
 POST /api/users/(str:username)/notifications/, 112
 PUT /api/users/(str:username)/, 111
 PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 113
 DELETE /api/users/(str:username)/, 112
 DELETE /api/users/(str:username)/groups/, 112
 DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 113
 PATCH /api/users/(str:username)/, 111
 PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 113

/exports

GET /exports/rss/, 155
 GET /exports/rss/(string:project)/, 155
 GET /exports/rss/(string:project)/(string:component)/, 155
 GET /exports/rss/(string:project)/(string:component)/(string:language)/, 155
 GET /exports/rss/language/(string:language)/, 155
 GET /exports/stats/(string:project)/(string:component)/, 154

/hooks

GET /hooks/update/(string:project)/, 151
 GET /hooks/update/(string:project)/(string:component)/, 151
 POST /hooks/azure/, 153
 POST /hooks/bitbucket/, 152
 POST /hooks/gitea/, 153
 POST /hooks/gitee/, 153
 POST /hooks/github/, 151

Sonderzeichen

.XML resource file
 file format, [91](#)

--add
 auto_translate command line option, [393](#)

--addon
 install_addon command line option, [400](#)

--age
 commit_pending command line option, [394](#)

--author
 add_suggestions command line option, [393](#)

--author-email
 wlc command line option, [159](#)

--author-name
 wlc command line option, [159](#)

--base-file-template
 import_project command line option, [398](#)

--check
 importusers command line option, [400](#)

--config
 wlc command line option, [157](#)

--config-section
 wlc command line option, [157](#)

--configuration
 install_addon command line option, [400](#)

--convert
 wlc command line option, [158](#)

--email
 createadmin command line option, [395](#)

--file-format
 import_project command line option, [398](#)

--force
 loadpo command line option, [401](#)

--force-commit
 pushgit command line option, [402](#)

--format
 wlc command line option, [157](#)

--fuzzy
 wlc command line option, [159](#)

--ignore
 import_json command line option, [396](#)

--inconsistent
 auto_translate command line option, [393](#)

--input
 wlc command line option, [158](#)

--key
 wlc command line option, [157](#)

--lang
 loadpo command line option, [401](#)

--language-code
 list_translators command line option, [401](#)

--language-map
 import_memory command line option, [397](#)

--language-regex
 import_project command line option, [398](#)

--license
 import_project command line option, [398](#)

--license-url
 import_project command line option, [398](#)

--main-component
 import_json command line option, [396](#)
 import_project command line option, [398](#)

--method
 wlc command line option, [159](#)

--mode
 auto_translate command line option, [394](#)

--mt
 auto_translate command line option, [393](#)

--name
 createadmin command line option, [395](#)

--name-template

- import_project command line option, 398
- new-base-template
 - import_project command line option, 398
- no-password
 - createadmin command line option, 395
- no-privs-update
 - setupgroups command line option, 403
- no-projects-update
 - setupgroups command line option, 403
- no-update
 - setuplang command line option, 403
- output
 - wlc command line option, 158
- overwrite
 - auto_translate command line option, 393
 - wlc command line option, 158
- password
 - createadmin command line option, 395
- project
 - import_json command line option, 396
- source
 - auto_translate command line option, 393
- threshold
 - auto_translate command line option, 394
- update
 - createadmin command line option, 395
 - import_json command line option, 396
 - install_addon command line option, 400
- url
 - wlc command line option, 157
- user
 - auto_translate command line option, 393
- username
 - createadmin command line option, 395
- vcs
 - import_project command line option, 398

A

- add_suggestions
 - weblate admin command, 393
- add_suggestions command line option
 - author, 393
- ADMINS
 - setting, 214
- AKISMET_API_KEY
 - setting, 349
- ALLOWED_HOSTS
 - setting, 214
- Android
 - file format, 85

- ANONYMOUS_USER_NAME
 - setting, 349
- API, 106, 155, 160
- Apple strings
 - file format, 86
- ARB
 - file format, 90
- AUDITLOG_EXPIRY
 - setting, 350
- AUTH_LOCK_ATTEMPTS
 - setting, 350
- AUTH_TOKEN_VALID
 - setting, 351
- auto_translate
 - weblate admin command, 393
- auto_translate command line option
 - add, 393
 - inconsistent, 393
 - mode, 394
 - mt, 393
 - overwrite, 393
 - source, 393
 - threshold, 394
 - user, 393

- AUTO_UPDATE
 - setting, 350

- AUTOFIX_LIST
 - setting, 351

- AVATAR_URL_PREFIX
 - setting, 351

B

- BACKGROUND_TASKS
 - setting, 352
- BaseAddon (*Klasse in weblate.addons.base*), 438
- BASIC_LANGUAGES
 - setting, 352
- bilingual
 - translation, 77
- BITBUCKETSERVER_CREDENTIALS
 - setting, 360
- BORG_EXTRA_ARGS
 - setting, 353

C

- CACHE_DIR
 - setting, 353
- can_install() (*Klassenmethode von weblate.addons.base.BaseAddon*), 438
- CELERY_BACKUP_OPTIONS, 168, 186
- CELERY_BEAT_OPTIONS, 168, 186
- CELERY_MAIN_OPTIONS, 168, 186
- CELERY_MEMORY_OPTIONS, 168, 186
- CELERY_NOTIFY_OPTIONS, 168, 186
- celery_queues
 - weblate admin command, 394
- CELERY_TRANSLATE_OPTIONS, 168, 186
- changes

- wlc command line option, 158
- CHECK_LIST
 - setting, 353
- checkgit
 - weblate admin command, 394
- cleanup
 - wlc command line option, 158
- cleanup_ssh_keys
 - weblate admin command, 395
- cleanuptrans
 - weblate admin command, 395
- Comma separated values
 - file format, 92
- Command (*Klasse in wlc.main*), 162
- COMMENT_CLEANUP_DAYS
 - setting, 354
- commit
 - wlc command line option, 157
- commit_pending
 - weblate admin command, 394
- commit_pending command line option
 - age, 394
- COMMIT_PENDING_HOURS
 - setting, 354
- commitgit
 - weblate admin command, 394
- configure() (*Methode von weblate.addons.base.BaseAddon*), 438
- CONTACT_FORM
 - setting, 354
- createadmin
 - weblate admin command, 395
- createadmin command line option
 - email, 395
 - name, 395
 - no-password, 395
 - password, 395
 - update, 395
 - username, 395
- CSP_CONNECT_SRC
 - setting, 353
- CSP_FONT_SRC
 - setting, 353
- CSP_IMG_SRC
 - setting, 353
- CSP_SCRIPT_SRC
 - setting, 353
- CSP_STYLE_SRC
 - setting, 353
- CSV
 - file format, 92

D

- daily() (*Methode von weblate.addons.base.BaseAddon*), 438
- DATA_DIR
 - setting, 355
- DATABASE_BACKUP

- setting, 355
- DATABASES
 - setting, 214
- DEBUG
 - setting, 215
- DEFAULT_ACCESS_CONTROL
 - setting, 356
- DEFAULT_ADD_MESSAGE
 - setting, 356
- DEFAULT_ADDON_MESSAGE
 - setting, 356
- DEFAULT_ADDONS
 - setting, 357
- DEFAULT_AUTO_WATCH
 - setting, 356
- DEFAULT_COMMIT_MESSAGE
 - setting, 356
- DEFAULT_COMMITTER_EMAIL
 - setting, 357
- DEFAULT_COMMITTER_NAME
 - setting, 357
- DEFAULT_DELETE_MESSAGE
 - setting, 356
- DEFAULT_FROM_EMAIL
 - setting, 215
- DEFAULT_LANGUAGE
 - setting, 357
- DEFAULT_MERGE_MESSAGE
 - setting, 356
- DEFAULT_MERGE_STYLE
 - setting, 358
- DEFAULT_PAGE_LIMIT
 - setting, 366
- DEFAULT_PULL_MESSAGE
 - setting, 358
- DEFAULT_RESTRICTED_COMPONENT
 - setting, 356
- DEFAULT_SHARED_TM
 - setting, 358
- DEFAULT_TRANSLATION_PROPAGATION
 - setting, 358
- download
 - wlc command line option, 158
- DTD
 - file format, 94
- dump_memory
 - weblate admin command, 396
- dumpuserdata
 - weblate admin command, 396

E

- ENABLE_AVATARS
 - setting, 358
- ENABLE_HOOKS
 - setting, 358
- ENABLE_HTTPS
 - setting, 359
- ENABLE_SHARING

setting, 359
 EXTRA_HTML_HEAD
 setting, 359

F

file format
 .XML resource file, 91
 Android, 85
 Apple strings, 86
 ARB, 90
 Comma separated values, 92
 CSV, 92
 DTD, 94
 gettext, 79
 go-i18n, 89
 gotext, 90
 GWT properties, 83
 i18next, 88
 INI translations, 84
 Java properties, 82
 Joomla translations, 85
 JSON, 87
 mi18n lang, 83
 PHP strings, 87
 PO, 79
 Qt, 85
 RC, 95
 ResourceDictionary, 92
 RESX, 91
 Ruby YAML, 93
 Ruby YAML Ain't Markup Language, 93
 string resources, 85
 TS, 85
 WPF, 92
 XLIFF, 81
 XML, 94
 YAML, 93
 YAML Ain't Markup Language, 93

G

get() (*Methode von wlc.Weblate*), 161
 get_add_form() (*Klassenmethode von weblate.addons.base.BaseAddon*), 438
 GET_HELP_URL
 setting, 359
 get_settings_form() (*Methode von weblate.addons.base.BaseAddon*), 438
 gettext
 file format, 79
 GITEA_CREDENTIALS
 setting, 359
 GITHUB_CREDENTIALS
 setting, 360
 GITLAB_CREDENTIALS
 setting, 360
 go-i18n
 file format, 89
 GOOGLE_ANALYTICS_ID

setting, 361
 gotext
 file format, 90
 GWT properties
 file format, 83

H

HIDE_REPO_CREDENTIALS
 setting, 361
 HIDE_VERSION
 setting, 361

I

i18next
 file format, 88
 import_demo
 weblate admin command, 396
 import_json
 weblate admin command, 396
 import_json command line option
 --ignore, 396
 --main-component, 396
 --project, 396
 --update, 396
 import_memory
 weblate admin command, 397
 import_memory command line option
 --language-map, 397
 import_project
 weblate admin command, 397
 import_project command line option
 --base-file-template, 398
 --file-format, 398
 --language-regex, 398
 --license, 398
 --license-url, 398
 --main-component, 398
 --name-template, 398
 --new-base-template, 398
 --vcs, 398
 importuserdata
 weblate admin command, 400
 importusers
 weblate admin command, 400
 importusers command line option
 --check, 400
 INI translations
 file format, 84
 install_addon
 weblate admin command, 400
 install_addon command line option
 --addon, 400
 --configuration, 400
 --update, 400
 INTERLEDGER_PAYMENT_POINTERS
 setting, 361
 iOS
 translation, 86

IP_BEHIND_REVERSE_PROXY
 setting, 362
 IP_PROXY_HEADER
 setting, 362
 IP_PROXY_OFFSET
 setting, 362

J

Java properties
 file format, 82
 Joomla translations
 file format, 85
 JSON
 file format, 87

L

LEGAL_TOS_DATE
 setting, 363
 LEGAL_URL
 setting, 363
 LICENSE_EXTRA
 setting, 363
 LICENSE_FILTER
 setting, 364
 LICENSE_REQUIRED
 setting, 364
 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
 setting, 364
 list_languages
 weblate admin command, 400
 list_translators
 weblate admin command, 401
 list_translators command line option
 --language-code, 401
 list_versions
 weblate admin command, 401
 list-components
 wlc command line option, 157
 list-languages
 wlc command line option, 157
 list-projects
 wlc command line option, 157
 list-translations
 wlc command line option, 157
 load() (*Methode von wlc.config.WeblateConfig*), 162
 loadpo
 weblate admin command, 401
 loadpo command line option
 --force, 401
 --lang, 401
 LOCALIZE_CDN_PATH
 setting, 364
 LOCALIZE_CDN_URL
 setting, 364
 lock
 wlc command line option, 158
 lock_translation
 weblate admin command, 401

lock-status
 wlc command line option, 158
 LOGIN_REQUIRED_URLS
 setting, 365
 LOGIN_REQUIRED_URLS_EXCEPTIONS
 setting, 365
 ls
 wlc command line option, 157

M

main() (*im Modul wlc.main*), 162
 MATOMO_SITE_ID
 setting, 365
 MATOMO_URL
 setting, 366
 mi18n lang
 file format, 83
 Modul
 wlc, 161
 wlc.config, 162
 wlc.main, 162
 monolingual
 translation, 77
 move_language
 weblate admin command, 402

N

NEARBY_MESSAGES
 setting, 366

P

PAGURE_CREDENTIALS
 setting, 366
 PHP strings
 file format, 87
 PIWIK_SITE_ID
 setting, 365
 PIWIK_URL
 setting, 366
 PO
 file format, 79
 post() (*Methode von wlc.Weblate*), 161
 post_add() (*Methode von weblate.addons.base.BaseAddon*), 438
 post_commit() (*Methode von weblate.addons.base.BaseAddon*), 438
 post_push() (*Methode von weblate.addons.base.BaseAddon*), 438
 post_update() (*Methode von weblate.addons.base.BaseAddon*), 438
 pre_commit() (*Methode von weblate.addons.base.BaseAddon*), 439
 pre_push() (*Methode von weblate.addons.base.BaseAddon*), 439
 pre_update() (*Methode von weblate.addons.base.BaseAddon*), 439
 PRIVACY_URL
 setting, 367

PRIVATE_COMMIT_EMAIL_OPT_IN
setting, 367

PRIVATE_COMMIT_EMAIL_TEMPLATE
setting, 367

PROJECT_BACKUP_KEEP_COUNT
setting, 367

PROJECT_BACKUP_KEEP_DAYS
setting, 367

PROJECT_NAME_RESTRICT_RE
setting, 368

PROJECT_WEB_RESTRICT_RE
setting, 368

pull
wlc command line option, 158

push
wlc command line option, 158

pushgit
weblate admin command, 402

pushgit command line option
--force-commit, 402

Python, 160

Python Enhancement Proposals
PEP 484, 434, 449

Q

Qt
file format, 85

R

RATELIMIT_ATTEMPTS
setting, 368

RATELIMIT_LOCKOUT
setting, 368

RATELIMIT_WINDOW
setting, 368

RC
file format, 95

REDIS_PASSWORD, 182

register_command() (*im Modul wlc.main*), 162

REGISTRATION_ALLOW_BACKENDS
setting, 369

REGISTRATION_CAPTCHA
setting, 369

REGISTRATION_EMAIL_MATCH
setting, 369

REGISTRATION_OPEN
setting, 370

REGISTRATION_REBIND
setting, 370

repo
wlc command line option, 158

REPOSITORY_ALERT_THRESHOLD
setting, 370

REQUIRE_LOGIN
setting, 370

reset
wlc command line option, 158

ResourceDictionary

file format, 92

REST, 106

RESX
file format, 91

RFC
RFC 5646, 76

Ruby YAML
file format, 93

Ruby YAML Ain't Markup Language
file format, 93

S

save_state() (*Methode von weblate.addons.base.BaseAddon*), 439

SECRET_KEY
setting, 215

SENTRY_DSN
setting, 371

SERVER_EMAIL
setting, 215

SESSION_COOKIE_AGE_AUTHENTICATED
setting, 371

SESSION_ENGINE
setting, 214

setting
ADMINS, 214
AKISMET_API_KEY, 349
ALLOWED_HOSTS, 214
ANONYMOUS_USER_NAME, 349
AUDITLOG_EXPIRY, 350
AUTH_LOCK_ATTEMPTS, 350
AUTH_TOKEN_VALID, 351
AUTO_UPDATE, 350
AUTOFIX_LIST, 351
AVATAR_URL_PREFIX, 351
BACKGROUND_TASKS, 352
BASIC_LANGUAGES, 352
BITBUCKETSERVER_CREDENTIALS, 360
BORG_EXTRA_ARGS, 353
CACHE_DIR, 353
CHECK_LIST, 353
COMMENT_CLEANUP_DAYS, 354
COMMIT_PENDING_HOURS, 354
CONTACT_FORM, 354
CSP_CONNECT_SRC, 353
CSP_FONT_SRC, 353
CSP_IMG_SRC, 353
CSP_SCRIPT_SRC, 353
CSP_STYLE_SRC, 353
DATA_DIR, 355
DATABASE_BACKUP, 355
DATABASES, 214
DEBUG, 215
DEFAULT_ACCESS_CONTROL, 356
DEFAULT_ADD_MESSAGE, 356
DEFAULT_ADDON_MESSAGE, 356
DEFAULT_ADDONS, 357
DEFAULT_AUTO_WATCH, 356

- DEFAULT_COMMIT_MESSAGE, 356
- DEFAULT_COMMITER_EMAIL, 357
- DEFAULT_COMMITER_NAME, 357
- DEFAULT_DELETE_MESSAGE, 356
- DEFAULT_FROM_EMAIL, 215
- DEFAULT_LANGUAGE, 357
- DEFAULT_MERGE_MESSAGE, 356
- DEFAULT_MERGE_STYLE, 358
- DEFAULT_PAGE_LIMIT, 366
- DEFAULT_PULL_MESSAGE, 358
- DEFAULT_RESTRICTED_COMPONENT, 356
- DEFAULT_SHARED_TM, 358
- DEFAULT_TRANSLATION_PROPAGATION, 358
- ENABLE_AVATARS, 358
- ENABLE_HOOKS, 358
- ENABLE_HTTPS, 359
- ENABLE_SHARING, 359
- EXTRA_HTML_HEAD, 359
- GET_HELP_URL, 359
- GITEA_CREDENTIALS, 359
- GITHUB_CREDENTIALS, 360
- GITLAB_CREDENTIALS, 360
- GOOGLE_ANALYTICS_ID, 361
- HIDE_REPO_CREDENTIALS, 361
- HIDE_VERSION, 361
- INTERLEDGER_PAYMENT_POINTERS, 361
- IP_BEHIND_REVERSE_PROXY, 362
- IP_PROXY_HEADER, 362
- IP_PROXY_OFFSET, 362
- LEGAL_TOS_DATE, 363
- LEGAL_URL, 363
- LICENSE_EXTRA, 363
- LICENSE_FILTER, 364
- LICENSE_REQUIRED, 364
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 364
- LOCALIZE_CDN_PATH, 364
- LOCALIZE_CDN_URL, 364
- LOGIN_REQUIRED_URLS, 365
- LOGIN_REQUIRED_URLS_EXCEPTIONS, 365
- MATOMO_SITE_ID, 365
- MATOMO_URL, 366
- NEARBY_MESSAGES, 366
- PAGURE_CREDENTIALS, 366
- PIWIK_SITE_ID, 365
- PIWIK_URL, 366
- PRIVACY_URL, 367
- PRIVATE_COMMIT_EMAIL_OPT_IN, 367
- PRIVATE_COMMIT_EMAIL_TEMPLATE, 367
- PROJECT_BACKUP_KEEP_COUNT, 367
- PROJECT_BACKUP_KEEP_DAYS, 367
- PROJECT_NAME_RESTRICT_RE, 368
- PROJECT_WEB_RESTRICT_RE, 368
- RATELIMIT_ATTEMPTS, 368
- RATELIMIT_LOCKOUT, 368
- RATELIMIT_WINDOW, 368
- REGISTRATION_ALLOW_BACKENDS, 369
- REGISTRATION_CAPTCHA, 369
- REGISTRATION_EMAIL_MATCH, 369
- REGISTRATION_OPEN, 370
- REGISTRATION_REBIND, 370
- REPOSITORY_ALERT_THRESHOLD, 370
- REQUIRE_LOGIN, 370
- SECRET_KEY, 215
- SENTRY_DSN, 371
- SERVER_EMAIL, 215
- SESSION_COOKIE_AGE_AUTHENTICATED, 371
- SESSION_ENGINE, 214
- SIMPLIFY_LANGUAGES, 371
- SINGLE_PROJECT, 372
- SITE_DOMAIN, 371
- SITE_TITLE, 372
- SPECIAL_CHARS, 372
- SSH_EXTRA_ARGS, 372
- STATUS_URL, 373
- SUGGESTION_CLEANUP_DAYS, 373
- UPDATE_LANGUAGES, 373
- URL_PREFIX, 373
- VCS_API_DELAY, 373
- VCS_BACKENDS, 374
- VCS_CLONE_DEPTH, 374
- WEBLATE_ADDONS, 374
- WEBLATE_EXPORTERS, 375
- WEBLATE_FORMATS, 375
- WEBLATE_GPG_IDENTITY, 376
- WEBLATE_MACHINERY, 376
- WEBSITE_REQUIRED, 376
- setupgroups
 - weblate admin command, 403
- setupgroups command line option
 - no-privs-update, 403
 - no-projects-update, 403
- setuplang
 - weblate admin command, 403
- setuplang command line option
 - no-update, 403
- show
 - wlc command line option, 157
- SIMPLIFY_LANGUAGES
 - setting, 371
- SINGLE_PROJECT
 - setting, 372
- SITE_DOMAIN
 - setting, 371
- SITE_TITLE
 - setting, 372
- SPECIAL_CHARS
 - setting, 372
- SSH_EXTRA_ARGS
 - setting, 372
- stats
 - wlc command line option, 158
- STATUS_URL
 - setting, 373

store_post_load() (*Methode von weblate.addons.base.BaseAddon*), 439

string resources
file format, 85

SUGGESTION_CLEANUP_DAYS
setting, 373

T

translation
bilingual, 77
iOS, 86
monolingual, 77

TS
file format, 85

U

Umgebungsvariable
CELERY_BACKUP_OPTIONS, 168, 186
CELERY_BEAT_OPTIONS, 168, 186
CELERY_MAIN_OPTIONS, 168, 186
CELERY_MEMORY_OPTIONS, 168, 186
CELERY_NOTIFY_OPTIONS, 168, 186
CELERY_TRANSLATE_OPTIONS, 168, 186
POSTGRES_ALTER_ROLE, 181
POSTGRES_CONN_MAX_AGE, 181
POSTGRES_DATABASE, 181
POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 181
POSTGRES_HOST, 181
POSTGRES_PASSWORD, 180
POSTGRES_PASSWORD_FILE, 180
POSTGRES_PORT, 181
POSTGRES_SSL_MODE, 181
POSTGRES_USER, 181
REDIS_DB, 182
REDIS_HOST, 182
REDIS_PASSWORD, 182
REDIS_PASSWORD_FILE, 182
REDIS_PORT, 182
REDIS_TLS, 182
REDIS_VERIFY_SSL, 182
ROLLBAR_ENVIRONMENT, 184
ROLLBAR_KEY, 184
SENTRY_DSN, 184
SENTRY_ENVIRONMENT, 184
SOCIAL_AUTH_SLACK_SECRET, 180
UWSGI_WORKERS, 168
WEB_WORKERS, 186
WEBLATE_ADD_ADDONS, 185
WEBLATE_ADD_APPS, 185
WEBLATE_ADD_AUTOFIX, 185
WEBLATE_ADD_CHECK, 185
WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 172
WEBLATE_ADMIN_EMAIL, 168--170
WEBLATE_ADMIN_NAME, 168--170
WEBLATE_ADMIN_PASSWORD, 164, 168--170, 204

WEBLATE_ADMIN_PASSWORD_FILE, 170
WEBLATE_AKISMET_API_KEY, 174, 410
WEBLATE_ALLOWED_HOSTS, 170, 214, 219, 372
WEBLATE_API_RATELIMIT_ANON, 109, 175
WEBLATE_API_RATELIMIT_USER, 109, 175
WEBLATE_AUTH_LDAP_BIND_DN, 176
WEBLATE_AUTH_LDAP_BIND_PASSWORD, 176
WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE, 176
WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS, 176
WEBLATE_AUTH_LDAP_SERVER_URI, 176
WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 176
WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 176
WEBLATE_AUTH_LDAP_USER_SEARCH, 176
WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 176
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 176
WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 176
WEBLATE_AUTO_UPDATE, 184
WEBLATE_AVATAR_URL_PREFIX, 175
WEBLATE_BASIC_LANGUAGES, 174
WEBLATE_BITBUCKETSERVER_HOST, 173
WEBLATE_BITBUCKETSERVER_TOKEN, 173
WEBLATE_BITBUCKETSERVER_USERNAME, 173
WEBLATE_BORG_EXTRA_ARGS, 175
WEBLATE_CONTACT_FORM, 170
WEBLATE_CORS_ALLOWED_ORIGINS, 176
WEBLATE_CSP_CONNECT_SRC, 174
WEBLATE_CSP_FONT_SRC, 174
WEBLATE_CSP_IMG_SRC, 174
WEBLATE_CSP_SCRIPT_SRC, 174
WEBLATE_CSP_STYLE_SRC, 174
WEBLATE_DATABASE_BACKUP, 182
WEBLATE_DEBUG, 169, 239
WEBLATE_DEFAULT_ACCESS_CONTROL, 173
WEBLATE_DEFAULT_AUTO_WATCH, 174
WEBLATE_DEFAULT_COMMITTER_EMAIL, 173
WEBLATE_DEFAULT_COMMITTER_NAME, 173
WEBLATE_DEFAULT_FROM_EMAIL, 170
WEBLATE_DEFAULT_PULL_MESSAGE, 173
WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 173
WEBLATE_DEFAULT_SHARED_TM, 174
WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 173
WEBLATE_EMAIL_BACKEND, 184
WEBLATE_EMAIL_HOST, 183
WEBLATE_EMAIL_HOST_PASSWORD, 183
WEBLATE_EMAIL_HOST_PASSWORD_FILE, 183

- WEBLATE_EMAIL_HOST_USER, 183
- WEBLATE_EMAIL_PORT, 183
- WEBLATE_EMAIL_USE_SSL, 183
- WEBLATE_EMAIL_USE_TLS, 183
- WEBLATE_ENABLE_AVATARS, 175
- WEBLATE_ENABLE_HOOKS, 175
- WEBLATE_ENABLE_HTTPS, 171, 250
- WEBLATE_ENABLE_SHARING, 175
- WEBLATE_EXTRA_HTML_HEAD, 175
- WEBLATE_GET_HELP_URL, 184
- WEBLATE_GITEA_HOST, 173
- WEBLATE_GITEA_TOKEN, 173
- WEBLATE_GITEA_USERNAME, 173
- WEBLATE_GITHUB_HOST, 172, 238
- WEBLATE_GITHUB_TOKEN, 172
- WEBLATE_GITHUB_USERNAME, 172
- WEBLATE_GITLAB_HOST, 172
- WEBLATE_GITLAB_TOKEN, 172
- WEBLATE_GITLAB_USERNAME, 172
- WEBLATE_GOOGLE_ANALYTICS_ID, 172
- WEBLATE_GPG_IDENTITY, 174
- WEBLATE_HIDE_VERSION, 174
- WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 171
- WEBLATE_IP_PROXY_HEADER, 171
- WEBLATE_LEGAL_URL, 184
- WEBLATE_LICENSE_FILTER, 174
- WEBLATE_LICENSE_REQUIRED, 174
- WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 175
- WEBLATE_LOCALIZE_CDN_PATH, 185
- WEBLATE_LOCALIZE_CDN_URL, 185
- WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 172
- WEBLATE_LOGLEVEL, 169
- WEBLATE_LOGLEVEL_DATABASE, 169
- WEBLATE_NO_EMAIL_AUTH, 180
- WEBLATE_PAGURE_HOST, 173
- WEBLATE_PAGURE_TOKEN, 173
- WEBLATE_PAGURE_USERNAME, 173
- WEBLATE_PRIVACY_URL, 184
- WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN, 176
- WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE, 175
- WEBLATE_RATELIMIT_ATTEMPTS, 174, 412
- WEBLATE_RATELIMIT_LOCKOUT, 174
- WEBLATE_RATELIMIT_WINDOW, 174
- WEBLATE_REGISTRATION_ALLOW_BACKENDS, 170
- WEBLATE_REGISTRATION_OPEN, 170
- WEBLATE_REGISTRATION_REBIND, 171
- WEBLATE_REMOVE_ADDONS, 185
- WEBLATE_REMOVE_APPS, 185
- WEBLATE_REMOVE_AUTOFIX, 185
- WEBLATE_REMOVE_CHECK, 185
- WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 172
- WEBLATE_REQUIRE_LOGIN, 172, 371
- WEBLATE_SAML_IDP_ENTITY_ID, 180
- WEBLATE_SAML_IDP_IMAGE, 180
- WEBLATE_SAML_IDP_TITLE, 180
- WEBLATE_SAML_IDP_URL, 180
- WEBLATE_SAML_IDP_X509CERT, 180
- WEBLATE_SECURE_PROXY_SSL_HEADER, 171, 172
- WEBLATE_SERVER_EMAIL, 170
- WEBLATE_SERVICE, 168, 186
- WEBLATE_SILENCED_SYSTEM_CHECKS, 174, 246
- WEBLATE_SIMPLIFY_LANGUAGES, 173
- WEBLATE_SITE_DOMAIN, 169, 216, 235, 372
- WEBLATE_SITE_TITLE, 169
- WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 179
- WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 179
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 179
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, 179
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID, 179
- WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 178
- WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY, 178
- WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET, 178
- WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 178
- WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 178
- WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 178
- WEBLATE_SOCIAL_AUTH_FEDORA, 179
- WEBLATE_SOCIAL_AUTH_GITEA_API_URL, 178
- WEBLATE_SOCIAL_AUTH_GITEA_KEY, 178
- WEBLATE_SOCIAL_AUTH_GITEA_SECRET, 178
- WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 177
- WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY, 177
- WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME, 177
- WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET, 177
- WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 177
- WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID, 177
- WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY, 177
- WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET, 177

- move_language, 402
 - pushgit, 402
 - setupgroups, 403
 - setuplang, 403
 - unlock_translation, 402
 - updatechecks, 403
 - updategit, 403
 - WEBLATE_ADDONS
 - setting, 374
 - WEBLATE_ADMIN_EMAIL, 168, 170
 - WEBLATE_ADMIN_NAME, 168, 170
 - WEBLATE_ADMIN_PASSWORD, 164, 168--170, 204
 - WEBLATE_ADMIN_PASSWORD_FILE, 170
 - WEBLATE_AKISMET_API_KEY, 410
 - WEBLATE_ALLOWED_HOSTS, 214, 219, 372
 - WEBLATE_API_RATELIMIT_ANON, 109
 - WEBLATE_API_RATELIMIT_USER, 109
 - WEBLATE_AUTH_LDAP_BIND_PASSWORD, 176
 - WEBLATE_DEBUG, 239
 - WEBLATE_EMAIL_HOST_PASSWORD, 183
 - WEBLATE_EMAIL_PORT, 183
 - WEBLATE_EMAIL_USE_SSL, 183
 - WEBLATE_EMAIL_USE_TLS, 183
 - WEBLATE_ENABLE_HTTPS, 250
 - WEBLATE_EXPORTERS
 - setting, 375
 - WEBLATE_FORMATS
 - setting, 375
 - WEBLATE_GITHUB_HOST, 238
 - WEBLATE_GPG_IDENTITY
 - setting, 376
 - WEBLATE_LOCALIZE_CDN_PATH, 185
 - WEBLATE_MACHINERY
 - setting, 376
 - WEBLATE_RATELIMIT_ATTEMPTS, 412
 - WEBLATE_REQUIRE_LOGIN, 371
 - WEBLATE_SECURE_PROXY_SSL_HEADER, 171
 - WEBLATE_SERVICE, 168
 - WEBLATE_SILENCED_SYSTEM_CHECKS, 246
 - WEBLATE_SITE_DOMAIN, 216, 235, 372
 - WEBLATE_WORKERS, 168, 186
 - WeblateConfig (*Klasse in wlc.config*), 162
 - WeblateException, 161
 - WEBSITE_REQUIRED
 - setting, 376
 - wlc, 155
 - Modul, 161
 - wlc command line option
 - author-email, 159
 - author-name, 159
 - config, 157
 - config-section, 157
 - convert, 158
 - format, 157
 - fuzzy, 159
 - input, 158
 - key, 157
 - method, 159
 - output, 158
 - overwrite, 158
 - url, 157
 - changes, 158
 - cleanup, 158
 - commit, 157
 - download, 158
 - list-components, 157
 - list-languages, 157
 - list-projects, 157
 - list-translations, 157
 - lock, 158
 - lock-status, 158
 - ls, 157
 - pull, 158
 - push, 158
 - repo, 158
 - reset, 158
 - show, 157
 - stats, 158
 - unlock, 158
 - upload, 158
 - version, 157
 - wlc.config
 - Modul, 162
 - wlc.main
 - Modul, 162
 - WPF
 - file format, 92
- ## X
- XLIFF
 - file format, 81
 - XML
 - file format, 94
- ## Y
- YAML
 - file format, 93
 - YAML Ain't Markup Language
 - file format, 93