



The Weblate Manual

Yayım 4.16

Michal Čihař

01 Mar 2023

1	Kullanıcı belgeleri	1
1.1	Weblate basics	1
1.2	Registration and user profile	1
1.3	Translating using Weblate	10
1.4	Downloading and uploading translations	20
1.5	Sözlük	23
1.6	Denetimler ve düzeltmeler	26
1.7	Arama	58
1.8	Çeviri iş akışları	63
1.9	Sık sorulan sorular	67
1.10	Desteklenen dosya biçimleri	75
1.11	Sürüm denetimi bütünleştirilmesi	97
1.12	Weblate REST API uygulaması	105
1.13	Weblate istemcisi	154
1.14	Weblate Python API	159
2	Yönetici belgeleri	161
2.1	Yapılandırma yönergeleri	161
2.2	Weblate dağıtımları	228
2.3	Upgrading Weblate	229
2.4	Weblate yedeğini alma ve taşıma	237
2.5	Kimlik doğrulama	245
2.6	Erişim denetimi	255
2.7	Çeviri projeleri	264
2.8	Dil tanımları	281
2.9	Sürekli yerleştirme	284
2.10	Çeviri lisanslarını belirlemek	293
2.11	Çeviri süreci	295
2.12	Denetimler ve düzeltmeler	301
2.13	Kendiliğinden önerileri yapılandırmak	311
2.14	Eklentiler	322
2.15	Çeviri belleği	340
2.16	Yapılandırma	342
2.17	Örnek yapılandırma	368
2.18	Yönetim komutları	384
2.19	Duyurular	395
2.20	Bileşen listeleri	398
2.21	İsteğe bağlı Weblate modülleri	399
2.22	Weblate platformunu özelleştirmek	404
2.23	Yönetim arayüzü	407
2.24	Weblate için destek almak	415

2.25	Yasal belgeler	418
3	Katılımcı belgeleri	420
3.1	Weblate için katkıda bulunmak	420
3.2	Weblate kodlarına katkıda bulunmak	422
3.3	Weblate kaynak kodları	426
3.4	Debugging Weblate	427
3.5	Weblate iç yapısı	429
3.6	Eklentileri geliştirmek	430
3.7	Weblate frontend	432
3.8	Weblate sorunlarını bildirmek	433
3.9	Weblate testsuite and continuous integration	433
3.10	Data schemas	435
3.11	Releasing Weblate	439
3.12	Güvenlik ve gizlilik	440
3.13	Weblate modüllerine katkıda bulunmak	441
3.14	Weblate hakkında	442
3.15	Lisans	442
4	Değişiklik geçmişi	443
4.1	Weblate 4.16	443
4.2	Weblate 4.15.2	443
4.3	Weblate 4.15.1	444
4.4	Weblate 4.15	444
4.5	Weblate 4.14.2	445
4.6	Weblate 4.14.1	445
4.7	Weblate 4.14	446
4.8	Weblate 4.13.1	446
4.9	Weblate 4.13	446
4.10	Weblate 4.12.2	447
4.11	Weblate 4.12.1	447
4.12	Weblate 4.12	448
4.13	Weblate 4.11.2	448
4.14	Weblate 4.11.1	448
4.15	Weblate 4.11	449
4.16	Weblate 4.10.1	450
4.17	Weblate 4.10	450
4.18	Weblate 4.9.1	450
4.19	Weblate 4.9	451
4.20	Weblate 4.8.1	451
4.21	Weblate 4.8	452
4.22	Weblate 4.7.2	452
4.23	Weblate 4.7.1	452
4.24	Weblate 4.7	453
4.25	Weblate 4.6.2	453
4.26	Weblate 4.6.1	454
4.27	Weblate 4.6	454
4.28	Weblate 4.5.3	455
4.29	Weblate 4.5.2	455
4.30	Weblate 4.5.1	455
4.31	Weblate 4.5	456
4.32	Weblate 4.4.2	456
4.33	Weblate 4.4.1	457
4.34	Weblate 4.4	457
4.35	Weblate 4.3.2	458
4.36	Weblate 4.3.1	458
4.37	Weblate 4.3	459
4.38	Weblate 4.2.2	460

4.39	Weblate 4.2.1	460
4.40	Weblate 4.2	460
4.41	Weblate 4.1.1	461
4.42	Weblate 4.1	461
4.43	Weblate 4.0.4	462
4.44	Weblate 4.0.3	463
4.45	Weblate 4.0.2	463
4.46	Weblate 4.0.1	463
4.47	Weblate 4.0	464
4.48	Weblate 3.x serisi	464
4.49	Weblate 2.x serisi	476
4.50	Weblate 1.x serisi	487
4.51	Weblate 0.x serisi	491
Python Modül Dizini		495
HTTP Routing Table		496
Dizin		499

1.1 Weblate basics

1.1.1 Proje ve bileşen yapısı

In Weblate translations are organized into projects and components. Each project can contain number of components and those contain translations into individual languages. The component corresponds to one translatable file (for example *GNU gettext* or *Android string resources*). The projects are there to help you organize component into logical sets (for example to group all translations used within one application).

Internally, each project has translations to common strings propagated across other components within it by default. This lightens the burden of repetitive and multi version translation. The translation propagation can be disabled per *Bileşen yapılandırması* using *Çevirilerin yayılmasını sağlamak* in case the translations should diverge.

Ayrıca bakınız:

`../devel/integration`

1.2 Registration and user profile

1.2.1 Hesap açılışı

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 Pano

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

2.5 sürümünde geldi.

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

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

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, a user profile section shows 'Your profile' and a row of tabs: 'Languages', 'Preferences' (which is active), 'Notifications', 'Account', 'Profile', 'Teams', 'Licenses', 'Audit log', and 'API access'. The 'Preferences' panel is open, showing several settings:

- ☐ Hide completed translations on the dashboard
- Translation editor mode**: A dropdown menu set to 'Full editor'.
- Zen editor mode**: A dropdown menu set to 'Top to bottom'.
- Number of nearby strings**: A text input field containing '15'. Below it, a note says '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, a note says '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, a note says '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, 'Watched translations' (which is selected) and 'Suggested translations'.

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 *Bileşen listeleri*.

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

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

1.2.3 Kullanıcı profili

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.

Not: All language selections only offer currently translated languages.

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

Diller

1.2.4 Arayüz dili

Kullanıcı arayüzünün görüntülenmesini istediğiniz dili seçin.

Çevrilen diller

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 translations 13		Suggested translations 5		Insights	Search		
Translation	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
WeblateOrg/Android — Czech MIT	76%	3	3	10			
WeblateOrg/Django — Czech GPL-3.0	96%	1	12	69	4		
WeblateOrg/Django — Hebrew GPL-3.0	92%	2	15	80			
WeblateOrg/Django — Hungarian GPL-3.0	69%	8	109	671	1		
WeblateOrg/Djangojs — Hungarian GPL-3.0	96%	2	6	28			
WeblateOrg/Djangojs — Hebrew GPL-3.0	✓						
WeblateOrg/Djangojs — Czech GPL-3.0	✓						
WeblateOrg/Language names — Czech GPL-3.0	✓						
WeblateOrg/Language names — Hungarian GPL-3.0	81%	4	5	32			
WeblateOrg/Language names — Hebrew GPL-3.0	✓						
WeblateOrg/WeblateOrg — Hungarian GPL-3.0	✓						
WeblateOrg/WeblateOrg — Czech GPL-3.0	✓						
WeblateOrg/WeblateOrg — Hebrew GPL-3.0	✓						

İkincil diller

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 this, the breadcrumb 'WeblateOrg / Django / Czech / Translate' is visible. A progress bar indicates 'translated 96%'. The main area shows a translation for the string 'Soubory'. It has input fields for Hebrew (קבצים), English (Files), and Czech (Soubory). Below the inputs are buttons: 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A table below shows 'Nearby strings' with 16 items, including Hungarian 'Fájlok' and English 'Files'. The right sidebar contains a 'Glossary' section and 'String information' with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

1.2.5 Ayarlar

Varsayılan pano görünümü

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.

Ayrıca bakınız:

Bileşen listeleri

Düzenleyici bağlantısı

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

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

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

Ayrıca bakınız:

You can find more info on registering custom URL protocols for the editor in the [Nette documentation](#).

Özel karakterler

Additional special characters to include in the *Sanal klavye*.

1.2.6 Bildirimler

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*.

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

İpucu: 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 Hesap

The *Account* tab lets you set up basic account details, connect various services you can use to sign in into Weblate, completely remove your account, or download your user data (see [Weblate kullanıcı verilerini dışa aktarma](#)).

Not: The list of services depends on your Weblate configuration, but can be made to include popular sites such as GitLab, GitHub, Google, Facebook, or Bitbucket or other OAuth 2.0 providers.

 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 Profil

Bu sayfadaki tüm alanlar isteğe bağlıdır ve istediğiniz zaman silinebilir. Bu bilgileri yazarak, bu verileri kullanıcı profilinizin görüldüğü her yerde paylaşmamıza izin vermiş olursunuz.

The commit e-mail will be used instead of your account e-mail in version control commits. Use this to avoid leaking your real e-mail there. Be aware that using different e-mail can disconnect your contributions on other servers (for example your contributions will no longer link to your profile on GitHub). The private e-mail can be turned on site-wide using `PRIVATE_COMMIT_EMAIL_OPT_IN`.

Avatar can be shown for each user (depending on `ENABLE_AVATARS`). These images are obtained using <https://gravatar.com/>.

1.2.9 Lisanslar

1.2.10 API erişimi

API erişim kodunuzu buradan alabilir ya da sıfırlayabilirsiniz.

1.2.11 Denetim günlüğü

Denetim günlüğü, hesabınızla yapılan işlemlerin kaydını tutar. Hesabınızla ilgili her önemli işlem için IP adresini ve kullanılan tarayıcıyı kaydeder. Kritik işlemler ayrıca birincil e-posta adresine bir bildirim gönderilmesini tetikler.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak

1.3 Translating using Weblate

Thank you for interest in translating using Weblate. Projects can either be set up for direct translation, or by way of accepting suggestions made by users without accounts.

Overall, there are two modes of translation:

- The project accepts direct translations
- The project only accepts suggestions, which are automatically validated once a defined number of votes is reached

Please see [Çeviri iş akışları](#) for more info on translation workflow.

Options for translation project visibility:

- Herkes görebilir
- Visible only to a certain group of translators

Ayrıca bakınız:

Erişim denetimi, Çeviri iş akışları

1.3.1 Çeviri projeleri

Translation projects hold related components; resources for the same software, book, or project.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the 'WeblateOrg' project is selected. The 'Components' tab is active, showing a table of translation components. The table has columns: Component, Translated, Unfinished, Unfinished words, Checks, Suggestions, and Comments. Three components are listed: 'Android' (79% translated, 30 unfinished words, 3 checks), 'Language names' (95% translated, 4 unfinished words, 5 checks), and 'Glossary' (100% translated, 0 unfinished words, 0 checks). A progress bar at the top right shows 'translated 85%'. A button 'Add new translation component' is visible at the bottom left.

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

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

1.3.2 Translation links

Having navigated to a component, a set of links lead to its actual translation. The translation is further divided into individual checks, like *Untranslated strings* or *Unfinished strings*. If the whole project is translated, without error, *All strings* is still available. Alternatively you can use the search field to find a specific string or term.

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / Django / Czech
translated 96%

Overview
Info
Search
Insights
Files
Tools
Manage
Share
Watching

Translation status

26 Strings 96%
185 Words 93%
Browse Translate

Strings status

26 All strings — 185 words Browse Translate Zen
25 Translated strings — 173 words Browse Translate Zen
1 Unfinished strings — 12 words Browse Translate Zen
1 Untranslated strings — 12 words Browse Translate Zen
1 Unfinished strings without suggestions — 12 words Browse Translate Zen
3 Strings with any failing checks — 11 words Browse Translate Zen
3 Translated strings with any failing checks — 11 words Browse Translate Zen
1 Failing check: Unchanged translation — 4 words Browse Translate Zen
1 Failing check: Mismatched full stop — 4 words Browse Translate Zen
1 Failing check: Python format — 3 words Browse Translate Zen

Other components

Component Translated Unfinished Unfinished words Unfinished characters Checks Suggestions Comments
Android 76% 3 3 10
Language names
Glossary WeblateOrg
Djangojs
Browse all components

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

1.3.3 Öneriler

Not: Actual permissions might vary depending on your Weblate configuration.

Anonymous users can only (by default) forward suggestions. Doing so is still available to signed-in users, in cases where uncertainty about the translation arises, prompting other translators to review it.

The suggestions are scanned on a daily basis to remove duplicates and suggestions matching the current translation.

1.3.4 Yorumlar

Three types of comments can be posted: for translations, source strings, or to report source string bugs when this functionality is turned on using *Kaynak onaylama kullanılsın*. Choose the one suitable to topic you want to discuss. Source string comments are in any event good for providing feedback on the original string, for example that it should be rephrased or to ask questions about it.

Tüm yorumlarda Markdown söz dizimi kullanılabilir ve @kullanıcı_adı yaparak diğer kullanıcıları anabilirsiniz.

Ayrıca bakınız:

report-source, *Kaynak dizge onayları*, *Kaynak onaylama kullanılsın*

1.3.5 Çeşitler

Variants are used to group different length variants of the string. The frontend of your project can then use different strings depending on the screen or window size.

Ayrıca bakınız:

variants, *Çeşitler*

1.3.6 Etiketler

Labels are used to categorize strings within a project to further customize the localization workflow (for example to define categories of strings).

Weblate tarafından şu etiketler kullanılır:

Kendiliğinden çevrilmiş

Dizge *Kendiliğinden çeviri* ile çevrilmiş.

Kaynağın onaylanması gerekli

Dizge *Kaynak dizge onayları* ile onaylanmak üzere imlenmiş.

Ayrıca bakınız:

labels

1.3.7 Çeviri

On the translation page, the source string and an editing area for its translation are shown. Should the translation be plural, multiple source strings and editing areas are shown, each described and labeled in the amount of plural forms the translated language has.

All special whitespace characters are underlined in red and indicated with grey symbols. More than one subsequent space is also underlined in red to alert the translator to a potential formatting issue.

Various bits of extra info can be shown on this page, most of which coming from the project source code (like context, comments or where the message is being used). Translation fields for any secondary languages translators select in the preferences will be shown (see *İkincil diller*) above the source string.

Below the translation, translators will find suggestion made by others, to be accepted (✓), accepted with changes (⇒), or deleted (🗑).

Çoğullar

Words changing form to account of their numeric designation are called plurals. Each language has its own definition of plurals. English, for example, supports one. In the singular definition of for example “car”, implicitly one car is referenced, in the plural definition, “cars” two or more cars are referenced (or the concept of cars as a noun). Languages like for example Czech or Arabic have more plurals and also their rules for plurals are different.

Weblate has full support for each of these forms, in each respective language (by translating every plural separately). The number of fields and how it is in turn used in the translated application or project depends on the configured plural formula. Weblate shows the basic info, and the [Language Plural Rules](#) by the Unicode Consortium is a more detailed description.

Ayrıca bakınız:

Çoğul formülü

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows the translation of the string '%(count)s word' from English to Czech. The English string is shown in the 'Singular' form, and the Czech translation is shown in the 'One' form. The interface includes a search bar, a 'Custom search' dropdown, and a 'Position and priority' dropdown. The right sidebar contains a 'Glossary' section, a 'String information' section, and a 'Comments' section. The 'String information' section includes fields for 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

Alternatif çeviriler

4.13 sürümünde geldi.

Not: This is currently only supported with *Çok değerli CSV dosyası*.

With some formats, it is possible to have more translations for a single string. You can add more alternative translations using the *Tools* menu. Any blank alternative translations will be automatically removed upon saving.

Klavye kısayolları

2.18 sürümünde değişti: Klavye kısayolları, 2.18 sürümünde tarayıcı veya sistem varsayılanlarıyla çakışma olasılığını azaltmak için yenilendi.

Çeviri sırasında şu klavye kısayolları kullanılabilir:

Klavye kısayolu	Açıklama
Alt+Home	Geçerli aramada ilk çeviriye gider.
Alt+End	Geçerli aramada son çeviriye gider.
Alt+PageUp ya da Ctrl+↑ ya da Alt+↑ ya da Cmd+↑	Geçerli aramada önceki çeviriye gider.
Alt+PageDown ya da Ctrl+↓ ya da Alt+↓ ya da Cmd+↓	Geçerli aramada sonraki çeviriye gider.
Ctrl+Enter ya da Cmd+Enter	Geçerli formu gönderir. Bu seçenek, çeviriye düzenlerken <i>Kaydet ve ilerile</i> düğmesine basmakla aynıdır.
Ctrl+Shift+Enter ya da Cmd+Shift+Enter	Düzenlenmesi gerekli imini kaldırır ve çeviriye gönderir.
Alt+Enter ya da Option+Enter	Dizgeyi öneri olarak gönderir. Bu seçenek, çeviriye düzenlerken <i>Suggest</i> düğmesine basmakla aynıdır.
Ctrl+E ya da Cmd+E	İmleci çeviri alanına taşır.
Ctrl+U ya da Cmd+U	İmleci yorum alanına taşır.
Ctrl+M ya da Cmd+M	<i>Kendiliğinden öneriler</i> sekmesini görüntüler. Bilgi almak için: <i>Kendiliğinden öneriler</i> .
Ctrl+1 to Ctrl+9 ya da Cmd+1 to Cmd+9	Kaynak dizgeden belirtilen numaralı yerleştirilebilir ögeyi kopyalar.
Ctrl+M+1 ile 9 arası ya da Cmd+M+1 ile 9 arası	Belirtilen numaralı makine çevirisini geçerli çeviriye kopyalar.
Ctrl+I+1 ile 9 arası ya da Cmd+I+1 ile 9 arası	Denetim hataları listesindeki bir ögeyi yok sayar.
Ctrl+J ya da Cmd+J	<i>Yakındaki dizgeler</i> sekmesini görüntüler.
Ctrl+S ya da Cmd+S	İmleci arama alanına taşır.
Ctrl+O ya da Cmd+O	Kaynak dizgeyi kopyalar.
Ctrl+Y ya da Cmd+Y	<i>Düzenlenmesi gerekli imini</i> değiştirir.

Sanal klavye

A small visual keyboard row is shown just above the translation field. This can be useful to keep local punctuation in mind (as the row is local to each language), or have characters otherwise hard to type handy.

The shown symbols factor into three categories:

- User configured *Özel karakterler* defined in the *Kullanıcı profili*
- Per-language characters provided by Weblate (e.g. quotes or RTL specific characters)
- Characters configured using *SPECIAL_CHARS*

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 / Hebrew / Translate' is visible. The main area is divided into a 'Translation' panel and a sidebar. The 'Translation' panel shows the 'English' source string 'Files' and the 'Hebrew' target string, which is currently empty. A visual keyboard is displayed above the target string input field. Below the input field, there are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. The sidebar on the right contains a 'Glossary' section with a search bar and a message 'No related strings found in the glossary.' Below that is the 'String information' section, which provides details about the current string, including its context, explanation, labels, flags, source string location, string age, source string age, and translation file.

Çeviri bağlamı

This contextual description provides related info about the current string.

Dizge öznitelikleri

Things like message ID, context (`msgctxt`) or location in source code.

Ekran görüntüleri

Screenshots can be uploaded to Weblate to better inform translators of where and how the string is used, see *Visual context for strings*.

Yakındaki dizgeler

Displays neighbouring messages from the translation file. These are usually also used in a similar context and prove useful in keeping the translation consistent.

Diğer kullanımlar

In case a message appears in multiple places (e.g. multiple components), this tab shows all of them if they are found to be inconsistent (see [Tutarsız](#)). You can choose which one to use.

Çeviri belleği

Look at similar strings translated in past, see [Çeviri belleği](#).

Sözlük

Displays terms from the project glossary used in the current message.

Son değişiklikler

List of people whom have changed this message recently using Weblate.

Proje

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.

Çeviri geçmişi

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.

Çevrilmiş dizgenin uzunluğu

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 [İki dilli ve tek dilli biçimler](#) for more info.
- Maximal length in characters defined by translation file or flag, see [Çevirinin en fazla uzunluğu](#).
- Maximal rendered size in pixels defined by flags, see [Çeviri en fazla uzunluğu](#).

1.3.8 Kendiliğinden öneriler

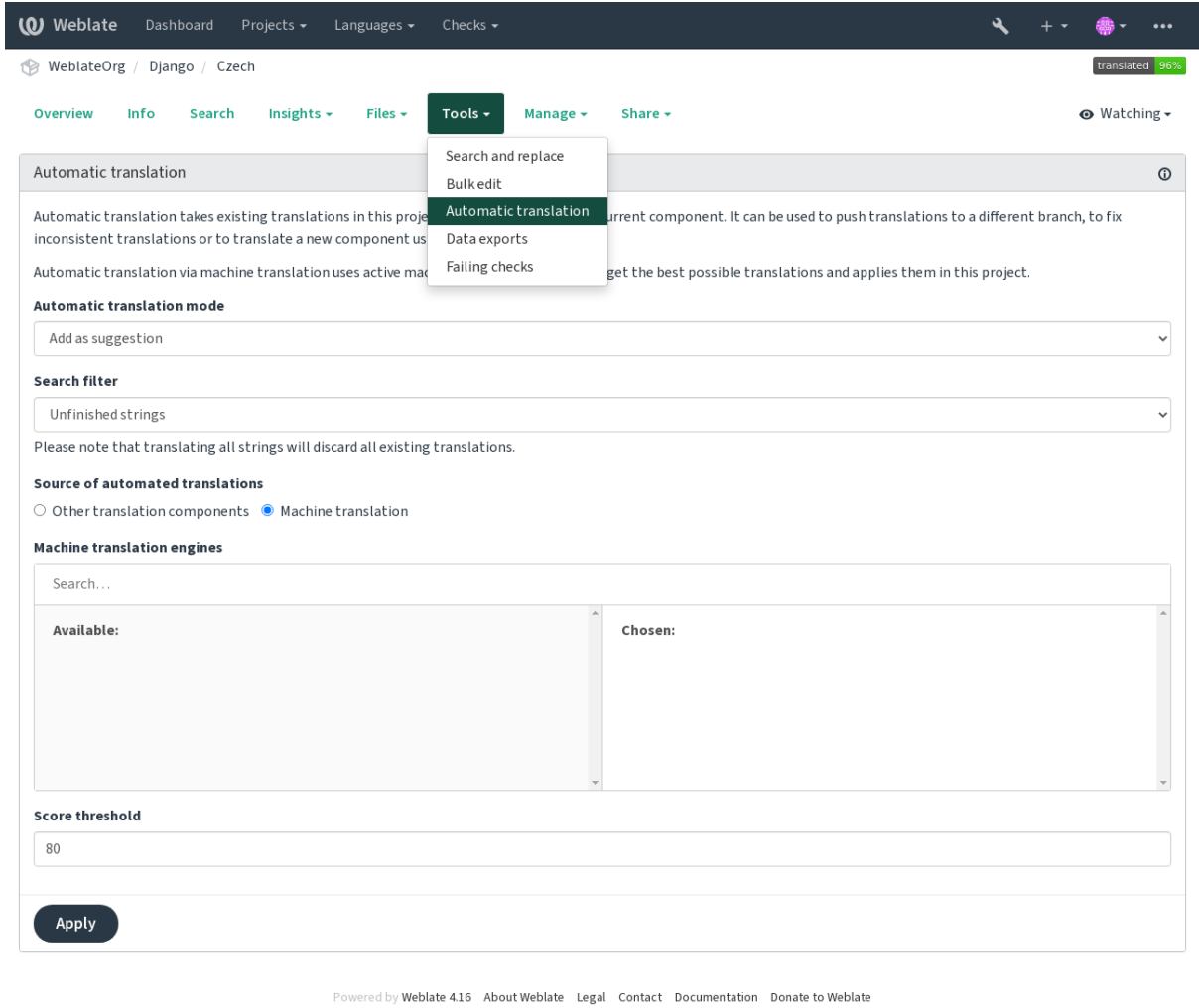
Based on configuration and your translated language, Weblate provides suggestions from several machine translation tools and [Çeviri belleği](#). All machine translations are available in a single tab of each translation page.

Ayrıca bakınız:

You can find the list of supported tools in [Kendiliğinden önerileri yapılandırmak](#).

1.3.9 Kendiliğinden çeviri

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:



İki şekilde işlem yapılabilir:

- Using other Weblate components as a source for translations.
- Using selected machine translation services with translations above a certain quality threshold.

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

Uyarı: 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).

Kendiliğinden çevrilmiş etiketinin görüntülenmesi sorunu düzeltildi.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

1.3.10 Hız sınırlama

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 [Hız sınırlama](#).

1.3.11 Arama ve değiştirme

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

İpucu: Dizgeleri karıştırma konusunda endişelenmeyin. Bu, asıl değişiklik onaylanmadan önce düzenlenen dizgelerin ön izlemesini gösteren iki adımlı bir işlemdir.

1.3.12 Toplu düzenleme

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 [Parametreleri kullanarak davranışı özelleştirmek](#))
- Adjust string labels (see labels)

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

Ayrıca bakınız:

[Bulk edit add-on](#)

1.3.13 Matris görünümü

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 kipi

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 *Ayarlar* tab on your *Kullanıcı profili*. Here you can also choose between having translations listed *Top to bottom* or *Side by side* depending on your personal preference.

1.4 Downloading and uploading translations

You can export files from a translation, make changes, and import them again. This allows working offline, and then merging changes back into the existing translation. This works even if it has been changed in the meantime.

Not: Available options might be limited by *access control* settings.

1.4.1 Downloading translations

From the project or component dashboard, translatable files can be downloaded in the *Files* menu.

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
- Gettext uzantılı XLIFF
- XLIFF 1.1
- TermBase eXchange
- Çeviri belleği takası
- gettext MO (yalnızca çeviri gettext PO kullanıyorsa kullanılabilir)
- CSV
- Excel Open XML
- JSON (yalnızca tek dilli çeviriler için kullanılabilir)
- Android dizge kaynağı (yalnızca tek dilli çeviriler için kullanılabilir)
- iOS dizgeleri (yalnızca tek dilli çeviriler için kullanılabilir)

İpucu: The content available in the converted files differs based on file format features, you can find overview in *Çeviri türü yetenekleri*.

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', and 'Upload translation'. The 'Customize download' dialog is displayed, showing a list of download options under 'Quick downloads' and a 'Customize download' section with a file format selector and a 'Download' button.

Quick downloads

Count	Description	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](#)

Ayrıca bakınız:

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

1.4.2 Uploading translations

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

Desteklenen dosya biçimleri

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.

Ayrıca bakınız:

Desteklenen dosya biçimleri, Downloading and uploading translations

Import methods

Çeviri dosyaları yüklenirken şu seçenekler sunulur:

Çeviri olarak eklensin (**translate**)

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

Only translations are used from the uploaded file and no additional content.

Öneri olarak eklensin (**suggest**)

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

Only translations are used from the uploaded file and no additional content.

Düzenlenmesi gerekli çeviri olarak eklensin (**fuzzy**)

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

Only translations are used from the uploaded file and no additional content.

Var olan çeviri dosyası değiştirilsin (replace)

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

Kaynak dizgeler güncellensin (source)

Updates source strings in bilingual translation file. This is similar to what *PO dosyalarını POT ile eşleştirerek güncelleme (msgmerge)* does.

Bu seçenek yalnızca bazı dosya biçimleri için kullanılabilir.

Yeni dizgeler eklensin (add)

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

In case you want to both add new strings and update existing translations, upload the file second time with *Add as translation*.

This option is available only with *Dizge yönetimi* turned on.

Yüklenen dosyadan yalnızca kaynak, çeviri ve anahtar (bağlam) kullanılır.

Ayrıca bakınız:

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

Conflicts handling

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

Düzenlenmesi gerekli dizgeler

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 Sözlük

Each project can include one or more glossaries as a shorthand for storing terminology. Glossary easify maintaining consistency of the translation.

A glossary for each language can be managed on its own, but they are stored together as a single component which helps project admins and multilingual translators to maintain some cross-language consistency as well. Terms from the glossary containing words from the currently translated string are displayed in the sidebar of the translation editor.

1.5.1 Managing glossaries

4.5 sürümünde değişti: Glossaries are now regular translation components and you can use all Weblate features on them — commenting, storing in a remote repository, or adding explanations.

Sözlük olarak kullanılsın ögesini açarak herhangi bir bileşeni sözlük olarak kullanabilirsiniz. Bir proje için birden çok sözlük oluşturabilirsiniz.

An empty glossary for a given project is automatically created with the project. Glossaries are shared among all components of the same project, and optionally with other projects using *Projelerde paylaşım* from the respective glossary component.

The glossary component looks like any other component in Weblate with added colored label:

Webblate Dashboard Projects Languages Checks

WebblateOrg / Glossary WebblateOrg / Czech translated 100%

Overview Info Search Insights Files Tools Share Not watching

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

Tüm sözlük terimlerine göz atın:

Webblate Dashboard Projects Languages Checks

WebblateOrg / Glossary WebblateOrg / Czech / Browse translated 100%

1/1 All strings Source string Add new glossary term

English	Czech
machine translation	strojový překlad
project	projekt

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

ya da bunları herhangi bir çeviri olarak düzenleyin.

1.5.2 Sözlük terimleri

Glossary terms are translated the same way regular strings are. You can toggle additional features using the *Tools* menu for each term.

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 trail reads 'WeblateOrg / Glossary / WeblateOrg / Czech / Translate'. A 'translated 100%' badge is visible. The main editing area is titled 'Glossary term' and contains fields for 'English' (project) and 'Czech' (projekt). There's a 'Needs editing' checkbox and an 'Explanation' field. At the bottom of the main area are buttons: 'Save and continue', 'Save and stay', 'Suggest', 'Skip', and 'Tools'. The 'Tools' menu is open, showing options: 'Delete string', 'Mark as untranslatable', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. On the right sidebar, the 'Glossary' section shows the current term and a '+ Add term to glossary' button. Below it, 'String information' shows 'String age' and 'Source string age' as 'a second ago', and 'Translation file' as 'cs.tbx, string 1 pending'. At the bottom, there's a 'Nearby strings' section with a table showing 'machine translation' and 'project' in English and Czech.

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

Çevrilemez terimler

4.5 sürümünde geldi.

Flagging certain glossary term translations `read-only` by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as untranslatable* means they can not be translated. Use this for brand names or other terms that should not be changed in other languages. Such terms are visually highlighted in the glossary sidebar.

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Yasaklı çeviriler

4.5 sürümünde geldi.

Flagging certain glossary term translations as `forbidden`, by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as forbidden translation* means they are **not** to be used. Use this to clarify translation when some words are ambiguous or could have unexpected meanings.

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Terminoloji

4.5 sürümünde geldi.

Flagging certain glossary terms as `terminology` by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as terminology* adds entries for them to all languages in the glossary. Use this for important terms that should be well thought out, and retain a consistent meaning across all languages.

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Çeşitler

Variants are a generic way to group strings together. All term variants are listed in the glossary sidebar when translating.

İpucu: You can use this to add abbreviations or shorter expressions for a term.

Ayrıca bakınız:

variants

1.6 Denetimler ve düzeltmeler

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:

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / 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
Zen

Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
Czech, Few
několiks 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
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. Denetimler ve düzeltmeler

27

1.6.1 Automatic fixups

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

Ayrıca bakınız:

AUTOFIX_LIST

1.6.2 Quality checks

Weblate employs a wide range of quality checks on strings. The following section describes them all in further detail. There are also language specific checks. Please file a bug if anything is reported in error.

Ayrıca bakınız:

CHECK_LIST, *Parametreleri kullanarak davranışı özelleştirmek*

1.6.3 Translation checks

Executed upon every translation change, helping translators maintain good quality translations.

BBCode kodlaması

Özet

Çevirideki BBCode kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.BBCodeCheck`

Belirteci denetle

`bbcode`

Yok sayılacak im

`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.

Not: The method for detecting BBCode is currently quite simple so this check might produce false positives.

Art arda yinelenen sözcükler

4.1 sürümünde geldi.

Özet

Metinde aynı sözcük üst üste iki kez geçiyor:

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.duplicate.DuplicateCheck`

Belirteci denetle

`duplicate`

Yok sayılacak im

ignore-duplicate

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

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

Sözlük ile uyumlu değil

4.5 sürümünde geldi.

Özet

Çeviri, bir sözlükte tanımlanmış terimlerle uyumlu değil.

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

weblate.checks.glossary.GlossaryCheck

Belirteci denetle

check_glossary

Etkinleştirilecek im

check-glossary

Yok sayılacak im

ignore-check-glossary

This check has to be turned on using `check-glossary` flag (see [Parametreleri kullanarak davranışı özelleştirmek](#)). 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.

Ayrıca bakınız:

[Sözlük](#), [Parametreleri kullanarak davranışı özelleştirmek](#), [Çeviri işaretleri](#)

Çift boşluk**Özet**

Çeviride çift boşluk var

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

weblate.checks.chars.DoubleSpaceCheck

Belirteci denetle

double_space

Yok sayılacak im

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.

Biçimlendirilmiş dizgeler

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 supports checking format strings in several languages. The check is not enabled automatically, only if a string is flagged appropriately (e.g. *c-format* for C format). Gettext adds this automatically, but you will probably have to add it manually for other file formats or if your PO files are not generated by **xgettext**.

This can be done per unit (see [Additional info on source strings](#)) or in *Bileşen yapılandırması*. 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.

İpucu: In case specific format check is not available in Weblate, you can use generic *Yer belirticiler*.

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

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area displays a translation unit for the string '%(count)s word'. The source string is in English and has singular and plural forms. The target string is in Czech and is currently empty. The interface includes a search bar, a sidebar with a glossary and string information, and a bottom section with a history of changes.

Translation Unit:

- 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:** ☐

Buttons: Save and continue, Save and stay, Suggest, Skip

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

History:

- None:** String updated in the repository
- WeblateOrg / Django — Czech:** 48 characters edited

Bottom section:

- English:**
 - Singular:** %(count)s word
 - Plural:** %(count)s words
 - Czech:**
 - One:** %(count)s slovo
 - Few:** %(count)s slova
 - Many:** %(count)s slov

AngularJS değer ekleme dizgesi

Özet

AngularJS değer ekleme dizgeleri kaynak ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.angularjs.AngularJSInterpolationCheck`

Belirteci denetle

`angularjs_format`

Etkinleştirilecek im

`angularjs-format`

Yok sayılacak im

`ignore-angularjs-format`

Adlandırılmış biçim dizgesi örneği

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

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, AngularJS metin değer ekleme

C biçimi

Özet

C biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.CFormatCheck`

Belirteci denetle

`c_format`

Etkinleştirilecek im

`c-format`

Yok sayılacak im

`ignore-c-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

Hesap bakiyeniz %1\$d %2\$s

Ayrıca bakınız:

Biçimlendirilmiş dizgeler,

C biçim dizgeleri, C printf biçimi

C# biçimi

Özet

C# biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.CSharpFormatCheck`

Belirteci denetle

`c_sharp_format`

Etkinleştirilecek im

`c-sharp-format`

Yok sayılacak im

`ignore-c-sharp-format`

Konum biçim dizgesi örneği

Sepette {0} elma var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [C# dizge biçimi](#)

ECMAScript kalıbı sabitleri

Özet

ECMAScript kalıbı sabitleri kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.ESTemplateLiteralsCheck`

Belirteci denetle

`es_format`

Etkinleştirilecek im

`es-format`

Yok sayılacak im

`ignore-es-format`

Değer ekleme örneği

Sepette \${number} elma var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Kalıp sabitleri](#)

i18next değer eklemesi

4.0 sürümünde geldi.

Özet

I18next değer eklemesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.I18NextInterpolationCheck`

Belirteci denetle

`i18next_interpolation`

Etkinleştirilecek im

`i18next-interpolation`

Yok sayılacak im

`ignore-i18next-interpolation`

Değer ekleme örneği

Sepette {{number}} elma var

İç içe geçme örneği

Sepette \$t(number) elma var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, i18next değer eklemesi

ICU MessageFormat

4.9 sürümünde geldi.

Özet

ICU MessageFormat dizgelerinde söz dizimi hataları ve/veya yer belirtici uyumsuzlukları var.

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.icu.ICUMessageFormatCheck`

Belirteci denetle

`icu_message_format`

Etkinleştirilecek im

`icu-message-format`

Yok sayılacak im

`ignore-icu-message-format`

Değer ekleme örneği

Sepette {number, plural, one {bir elma} other {# elma var}}.

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>	Düzenleyicide yer belirtici ögelerin vurgulanmasını kapatır.
<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>	Yer belirtici türlerinin kaynak ile uygunluğu denetimini atlar.
<code>-extra</code>	Skip checking that no placeholders are present that were not present in the source string.
<code>-missing</code>	Skip checking that no placeholders are missing that were present in the source string.

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`.

Ayrıca bakınız:

ICU MessageFormat söz dizimi, *Biçimlendirilmiş dizgeler*, *ICU: Formatting Messages*, *Format.JS: Message Syntax*

Java biçimi

Özet

Java biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.JavaFormatCheck`

Belirteci denetle

`java_printf_format`

Etkinleştirilecek im

`java-printf-format`

Yok sayılacak im

`ignore-java-printf-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

Hesap bakiyeniz %1\$d %2\$s

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

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, *Java biçim dizgesi*

Java MessageFormat

Özet

Java MessageFormat dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.JavaMessageFormatCheck`

Belirteci denetle

`java_format`

Koşulsuz olarak etkinleştirilecek im

`java-format`

Kendiliğinden algılamayı etkinleştirecek im

`auto-java-messageformat` enables check only if there is a format string in the source

Yok sayılacak im

`ignore-java-format`

Konum biçim dizgesi örneği

Sepette {0} elma var

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

This check validates that format string is valid for the Java MessageFormat class. Besides matching format strings in the curly braces, it also verifies single quotes as they have a special meaning. Whenever writing single quote, it should be written as `' '`. When not paired, it is treated as beginning of quoting and will not be shown when rendering the string.

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, Java Message biçimi

JavaScript biçimi

Özet

JavaScript biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.JavaScriptFormatCheck`

Belirteci denetle

`javascript_format`

Etkinleştirilecek im

`javascript-format`

Yok sayılacak im

`ignore-javascript-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, JavaScript dizge biçimi

Lua biçimi

Özet

Lua biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.LuaFormatCheck`

Belirteci denetle

`lua_format`

Etkinleştirilecek im

`lua-format`

Yok sayılacak im

`ignore-lua-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), *[Lua dizge biçimi](#)*

Object Pascal biçimi

Özet

Object Pascal biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.ObjectPascalFormatCheck`

Belirteci denetle

`object_pascal_format`

Etkinleştirilecek im

`object-pascal-format`

Yok sayılacak im

`ignore-object-pascal-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), *[Object Pascal biçimlendirme dizgeleri](#)*, *[Free Pascal biçimlendirme dizgeleri](#)* *[Delphi biçimlendirme dizgeleri](#)*

Yüzde yer belirticileri

4.0 sürümünde geldi.

Özet

Yüzde yer belirticileri kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PercentPlaceholdersCheck`

Belirteci denetle

`percent_placeholders`

Etkinleştirilecek im

`percent-placeholders`

Yok sayılacak im

`ignore-percent-placeholders`

Basit biçim dizgesi örneği

`%number% elma var`

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#),

Perl biçimi

Özet

Perl biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PerlFormatCheck`

Belirteci denetle

`perl_format`

Etkinleştirilecek im

`perl-format`

Yok sayılacak im

`ignore-perl-format`

Basit biçim dizgesi örneği

`Sepette %d elma var`

Konum biçim dizgesi örneği

`Hesap bakiyeniz %1$d %2$s`

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Perl sprintf](#), [Perl biçim dizgeleri](#)

PHP biçimi

Özet

PHP biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PHPFormatCheck`

Belirteci denetle

`php_format`

Etkinleştirilecek im

`php-format`

Yok sayılacak im

`ignore-php-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

Hesap bakiyeniz %1\$d %2\$s

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [PHP sprintf belgeleri](#), [PHP biçim dizgeleri](#)

Python parantez biçimi

Özet

Python ayracı biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PythonBraceFormatCheck`

Belirteci denetle

`python_brace_format`

Etkinleştirilecek im

`python-brace-format`

Yok sayılacak im

`ignore-python-brace-format`

Basit biçim dizgesi

`{}` elma var

Adlandırılmış biçim dizgesi örneği

Your balance is {amount} {currency}

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Python parantez biçimi](#), [Python biçim dizgeleri](#)

Python biçimi

Özet

Python biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.PythonFormatCheck`

Belirteci denetle

`python_format`

Etkinleştirilecek im

`python-format`

Yok sayılacak im

`ignore-python-format`

Basit biçim dizgesi

Sepette %d elma var

Adlandırılmış biçim dizgesi örneği

Your balance is %(amount)d %(currency)s

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Python dizge biçimlendirmeleri](#), [Python biçim dizgeleri](#)

Qt biçimi

Özet

Qt biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.qt.QtFormatCheck`

Belirteci denetle

`qt_format`

Etkinleştirilecek im

`qt-format`

Yok sayılacak im

`ignore-qt-format`

Konum biçim dizgesi örneği

There are %1 apples

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Qt QString::arg\(\)](#)

Qt çoğul biçimi

Özet

Qt çoğul biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.qt.QtPluralCheck`

Belirteci denetle

`qt_plural_format`

Etkinleştirilecek im

`qt-plural-format`

Yok sayılacak im

`ignore-qt-plural-format`

Çoğul biçim dizgesi örneği

There are %Ln apple(s)

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Qt 118n guide](#)

Ruby biçimi

Özet

Ruby biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.ruby.RubyFormatCheck`

Belirteci denetle

`ruby_format`

Etkinleştirilecek im

`ruby-format`

Yok sayılacak im

`ignore-ruby-format`

Basit biçim dizgesi örneği

Sepette %d elma var

Konum biçim dizgesi örneği

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

Adlandırılmış biçim dizgesi örneği

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

Named template string

Your balance is %{amount} %{currency}

Ayrıca bakınız:

[Biçimlendirilmiş dizgeler](#), [Ruby Kernel#sprintf](#)

Şema biçimi

Özet

Şema biçim dizgesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.SchemeFormatCheck`

Belirteci denetle

`scheme_format`

Etkinleştirilecek im

`scheme-format`

Yok sayılacak im

`ignore-scheme-format`

Basit biçim dizgesi örneği

There are ~d apples

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, Srfi 28, Chicken Scheme format, Guile Scheme formatted output

Vue I18n biçimlendirmesi

Özet

Vue I18n biçimlendirmesi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.format.VueFormattingCheck`

Belirteci denetle

`vue_format`

Etkinleştirilecek im

`vue-format`

Yok sayılacak im

`ignore-vue-format`

Adlandırılmış biçimlendirme

There are {count} apples

Rails i18n biçimlendirmesi

There are %{count} apples

Bağlantılı yerel iletiler

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

Ayrıca bakınız:

Biçimlendirilmiş dizgeler, Vue I18n Formatting, Vue I18n Linked locale messages

Çevrilmiş

Özet

Bu dizge geçmişte çevrilmiş

Kapsam

tüm dizgeler

Denetim sınıfı

`weblate.checks.consistency.TranslatedCheck`

Belirteci denetle

`translated`

Yok sayılacak im

`ignore-translated`

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

Tutarsız

Özet

Bu dizge bu projede farklı şekillerde çevrilmiş ya da bazı bileşenlerde çevrilmemiş.

Kapsam

tüm dizgeler

Denetim sınıfı

`weblate.checks.consistency.ConsistencyCheck`

Belirteci denetle

`inconsistent`

Yok sayılacak im

`ignore-inconsistent`

Weblate checks translations of the same string across all translation within a project to help you keep consistent translations.

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 *Çevirilerin yayılmasını sağlamak* turned on.

İpucu: For performance reasons, the check might not find all inconsistencies, it limits number of matches.

Not: 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 *Kendiliğinden çeviri* add-on to automate translating of newly added strings which are already translated in another component.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

Kashida harfi kullanılmış

3.5 sürümünde geldi.

Özet

Dekoratif kashida harfleri kullanılmamalıdır

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.KashidaCheck`

Belirteci denetle

`kashida`

Yok sayılacak im

`ignore-kashida`

The decorative Kashida letters should not be used in translation. These are also known as Tatweel.

Ayrıca bakınız:

[Kashida on Wikipedia](#)

Markdown bağlantıları

3.5 sürümünde geldi.

Özet

Markdown bağlantıları kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.MarkdownLinkCheck`

Belirteci denetle

`md-link`

Etkinleştirilecek im

`md-text`

Yok sayılacak im

`ignore-md-link`

Markdown links do not match source.

Ayrıca bakınız:

[Markdown links](#)

Markdown başvuruları

3.5 sürümünde geldi.

Özet

Markdown bağlantı başvuruları kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.MarkdownRefLinkCheck`

Belirteci denetle

md-reflink

Etkinleştirilecek im

md-text

Yok sayılacak im

ignore-md-reflink

Markdown link references do not match source.

Ayrıca bakınız:

[Markdown links](#)

Markdown söz dizimi

3.5 sürümünde geldi.

Özet

Markdown söz dizimi kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.MarkdownSyntaxCheck`

Belirteci denetle

md-syntax

Etkinleştirilecek im

md-text

Yok sayılacak im

ignore-md-syntax

Markdown söz dizimi kaynaktaki ile aynı değil

Ayrıca bakınız:

[Markdown span elements](#)

Çevirinin en fazla uzunluğu

Özet

Çeviri belirtilen uzunluğu aşmamalı

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.MaxLengthCheck`

Belirteci denetle

max-length

Etkinleştirilecek im

max-length

Yok sayılacak im

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`.

İpucu: This check looks at number of chars, what might not be the best metric when using proportional fonts to render the text. The [Çeviri en fazla uzunluğu](#) 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.

Çeviri en fazla uzunluğu

Özet

Çevrilmiş metin belirtilen boyutu aşmamalıdır

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.render.MaxSizeCheck`

Belirteci denetle

`max-size`

Etkinleştirilecek im

`max-size`

Yok sayılacak im

`ignore-max-size`

3.7 sürümünde geldi.

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 [Parametreleri kullanarak davranışı özelleştirmek](#)), 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
```

İpucu: You might want to set `font-*` directives in [Bileşen yapılandırması](#) 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.

Ayrıca bakınız:

[Yazı tiplerini yönetmek](#), [Parametreleri kullanarak davranışı özelleştirmek](#), [Çevirinin en fazla uzunluğu](#)

\n aynı değil

Özet

Çevirideki\n sabit yeni satır sayısı kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EscapedNewlineCountingCheck`

Belirteci denetle

`escaped_newline`

Yok sayılacak im

`ignore-escaped-newline`

Usually escaped newlines are important for formatting program output. Check fails if the number of \n literals in translation do not match the source.

İki nokta üst üste farklı

Özet

Kaynağın ve çevirinin ikisi de iki nokta üst üste ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndColonCheck`

Belirteci denetle

`end_colon`

Yok sayılacak im

`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).

Ayrıca bakınız:

[Colon on Wikipedia](#)

Üç nokta farklı

Özet

Kaynağın ve çevirinin ikisi de üç nokta ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndEllipsisCheck`

Belirteci denetle

`end_ellipsis`

Yok sayılacak im

`ignore-end-ellipsis`

Checks that trailing ellipses are replicated between both source and translation. This only checks for real ellipsis (...) not for three dots (. . .).

An ellipsis is usually rendered nicer than three dots in print, and sounds better with text-to-speech.

Ayrıca bakınız:[Ellipsis on Wikipedia](#)**Ünlem işareti farklı****Özet**

Kaynağın ve çevirinin ikisi de ünlem işareti ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndExclamationCheck`

Belirteci denetle

`end_exclamation`

Yok sayılacak im

`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).

Ayrıca bakınız:[Exclamation mark on Wikipedia](#)**Nokta işareti farklı****Özet**

Kaynağın ve çevirinin ikisi de nokta ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndStopCheck`

Belirteci denetle

`end_stop`

Yok sayılacak im

`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).

Ayrıca bakınız:[Full stop on Wikipedia](#)

Soru işareti farklı

Özet

Kaynağın ve çevirinin ikisi de soru işareti ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndQuestionCheck`

Belirteci denetle

`end_question`

Yok sayılacak im

`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).

Ayrıca bakınız:

[Question mark on Wikipedia](#)

Noktalı virgül farklı

Özet

Kaynağın ve çevirinin ikisi de noktalı virgül ile bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndSemicolonCheck`

Belirteci denetle

`end_semicolon`

Yok sayılacak im

`ignore-end-semicolon`

Checks that semicolons at the end of sentences are replicated between both source and translation.

Ayrıca bakınız:

[Semicolon on Wikipedia](#)

Satır sonları farklı

Özet

Çevirideki yeni satır sayısı kaynaktaki ile aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.NewLineCountCheck`

Belirteci denetle

`newline-count`

Yok sayılacak im

`ignore-newline-count`

Usually newlines are important for formatting program output. Check fails if the number of `\n` literals in translation do not match the source.

Çoğul biçim eksik

Özet

Bazı çoğul biçimler çevrilmemiş

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.consistency.PluralsCheck`

Belirteci denetle

`plurals`

Yok sayılacak im

`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.

Yer belirteciler

3.9 sürümünde geldi.

Özet

Çeviride bazı yer belirteciler eksik

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.placeholders.PlaceholderCheck`

Belirteci denetle

`placeholders`

Etkinleştirilecek im

`placeholders`

Yok sayılacak im

`ignore-placeholders`

4.3 sürümünde değişti: Kurallı ifadeyi yer belirtici olarak kullanabilirsiniz.

4.13 sürümünde değişti: 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
```

Ayrıca bakınız:

Parametreleri kullanarak davranışı özelleştirmek

Noktalama aralığı

3.9 sürümünde geldi.

Özet

Çift noktalama işaretinin önündeki birleşik boşluk eksik

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.PunctuationSpacingCheck`

Belirteci denetle

`punctuation_spacing`

Yok sayılacak im

`ignore-punctuation-spacing`

Checks that there is non breakable space before double punctuation sign (exclamation mark, question mark, semicolon and colon). This rule is used only in a few selected languages like French or Breton, where space before double punctuation sign is a typographic rule.

Ayrıca bakınız:

[French and English spacing on Wikipedia](#)

Kurallı ifade

3.9 sürümünde geldi.

Özet

Çevirideki kurallı ifade aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.placeholders.RegexCheck`

Belirteci denetle

`regex`

Etkinleştirilecek im

`regex`

Yok sayılacak im

`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$
```

Tekil ile çoğul biçim aynı

Özet

Bazı çoğul biçimler tekil biçimler ile aynı

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.consistency.SamePluralsCheck`

Belirteci denetle

`same-plurals`

Yok sayılacak im

`ignore-same-plurals`

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

Başta yeni satır var

Özet

Kaynağın ve çevirinin ikisi de yeni satırla başlamıyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.BeginNewlineCheck`

Belirteci denetle

`begin_newline`

Yok sayılacak im

`ignore-begin-newline`

Newlines usually appear in source strings for good reason, omissions or additions can lead to formatting problems when the translated text is put to use.

Ayrıca bakınız:

Sonda yeni satır var

Baştaki boşluk

Özet

Kaynağın ve çevirinin ikisi de aynı sayıda boşlukla başlamıyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.BeginSpaceCheck`

Belirteci denetle

`begin_space`

Yok sayılacak im

`ignore-begin-space`

A space in the beginning of a string is usually used for indentation in the interface and thus important to keep.

Sonda yeni satır var

Özet

Kaynağın ve çevirinin ikisi de yeni satırla bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndNewlineCheck`

Belirteci denetle

`end_newline`

Yok sayılacak im

`ignore-end-newline`

Newlines usually appear in source strings for good reason, omissions or additions can lead to formatting problems when the translated text is put to use.

Ayrıca bakınız:

Başta yeni satır var

Sondaki boşluk

Özet

Kaynağın ve çevirinin ikisi de boşlukla bitmiyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.EndSpaceCheck`

Belirteci denetle

`end_space`

Yok sayılacak im

`ignore-end-space`

Checks that trailing spaces are replicated between both source and translation.

Trailing space is usually utilized to space out neighbouring elements, so removing it might break layout.

Çeviri değiştirilmemiş

Özet

Kaynak ile çeviri aynı

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.same.SameCheck`

Belirteci denetle

`same`

Yok sayılacak im

`ignore-same`

Happens if the source and corresponding translation strings is identical, down to at least one of the plural forms. Some strings commonly found across all languages are ignored, and various markup is stripped. This reduces the number of false positives.

This check can help find strings mistakenly untranslated.

The default behavior of this check is to exclude words from the built-in blacklist from the checking. These are words which are frequently not being translated. This is useful to avoid false positives on short strings, which consist only of single word which is same in several languages. This blacklist can be disabled by adding `strict-same` flag to string or component.

Ayrıca bakınız:

Bileşen yapılandırması, Parametreleri kullanarak davranışı özelleştirmek

Güvenli olmayan HTML

3.9 sürümünde geldi.

Özet

Çeviride güvenli olmayan HTML kodlaması var

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.SafeHTMLCheck`

Belirteci denetle

`safe-html`

Etkinleştirilecek im

`safe-html`

Yok sayılacak im

`ignore-safe-html`

The translation uses unsafe HTML markup. This check has to be enabled using `safe-html` flag (see *Parametreleri kullanarak davranışı özelleştirmek*). There is also accompanied autofixer which can automatically sanitize the markup.

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

Ayrıca bakınız:

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

Adres

3.5 sürümünde geldi.

Özet

Çeviride bir adres bulunmuyor

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.URLCheck`

Belirteci denetle

`url`

Etkinleştirilecek im

`url`

Yok sayılacak im

`ignore-url`

The translation does not contain an URL. This is triggered only in case the unit is marked as containing URL. In that case the translation has to be a valid URL.

XML kodlaması

Özet

Çevirideki XML kod imleri kaynakla aynı değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.XMLTagsCheck`

Belirteci denetle

`xml-tags`

Yok sayılacak im

`ignore-xml-tags`

This usually means the resulting output will look different. In most cases this is not a desired result from changing the translation, but occasionally it is.

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

Not: This check is disabled by the `safe-html` flag as the HTML cleanup done by it can produce HTML markup which is not valid XML.

XML söz dizimi

2.8 sürümünde geldi.

Özet

Çeviri geçerli bir XML kodu değil

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.markup.XMLValidityCheck`

Belirteci denetle

`xml-invalid`

Yok sayılacak im

`ignore-xml-invalid`

The XML markup is not valid.

Not: This check is disabled by the `safe-html` flag as the HTML cleanup done by it can produce HTML markup which is not valid XML.

Sıfır genişlikli boşluk

Özet

Çeviride fazladan sıfır genişlikli boşluk karakteri var

Kapsam

çevrilmiş dizgeler

Denetim sınıfı

`weblate.checks.chars.ZeroWidthSpaceCheck`

Belirteci denetle

`zero-width-space`

Yok sayılacak im

`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.

Ayrıca bakınız:

[Zero width space on Wikipedia](#)

1.6.4 Kaynak denetimleri

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

Üç nokta

Özet

Dizge, üç nokta karakteri (...) yerine üç tane nokta (...) kullanıyor

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.EllipsisCheck`

Belirteci denetle

`ellipsis`

Yok sayılacak im

`ignore-ellipsis`

This fails when the string uses three dots (. . .) when it should use an ellipsis character (...).

Using the Unicode character is in most cases the better approach and looks better rendered, and may sound better with text-to-speech.

Ayrıca bakınız:

[Ellipsis on Wikipedia](#)

ICU MessageFormat söz dizimi

4.9 sürümünde geldi.

Özet

ICU MessageFormat dizgelerinde söz dizimi hataları var.

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.icu.ICUSourceCheck`

Belirteci denetle

`icu_message_format_syntax`

Etkinleştirilecek im

`icu-message-format`

Yok sayılacak im

`ignore-icu-message-format`

Ayrıca bakınız:

ICU MessageFormat

Uzun süredir çevrilmemiş

4.1 sürümünde geldi.

Özet

Dizge uzun zamandır çevrilmemiş

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.LongUntranslatedCheck`

Belirteci denetle

`long_untranslated`

Yok sayılacak im

`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.

Birden çok denetim hatası

Özet

Birkaç dildeki çevirilerde denetim hataları var

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.MultipleFailingCheck`

Belirteci denetle

`multiple_failures`

Yok sayılacak im

`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.

Birden çok değişken adlandırılmamış

4.1 sürümünde geldi.

Özet

Dizgede birden çok adlandırılmamış değişken var. Çevirmenler istese de bunların sıralamasını değiştiremez

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.format.MultipleUnnamedFormatsCheck`

Belirteci denetle

`unnamed_format`

Yok sayılacak im

`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.

Çoğul değil

Özet

Dizge çoğul olarak kullanılmış. Ancak çoğul biçimler kullanılmamış

Kapsam

kaynak dizgeler

Denetim sınıfı

`weblate.checks.source.OptionalPluralCheck`

Belirteci denetle

`optional_plural`

Yok sayılacak im

`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 Arama

3.9 sürümünde geldi.

İstedığınız dizgeleri bulmak için boole işlemleri, parantezler veya alana özgü arama kullanan gelişmiş sorgular kullanılabilir.

Herhangi bir alan belirtilmediğinde, arama kaynak, hedef ve bağlam dizgelerinde yapılır.

The screenshot shows the Weblate Search interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, a 'Dashboard' section shows 'Watched translations' (0), 'Suggested translations' (0), and 'Insights'. A 'Search' button is visible. The main search area has a 'Search' input field, a 'Sort By' dropdown, and a 'Filter' icon. Below the search bar is the 'Advanced query builder' section with 'Source strings', 'Search for...', 'Exact' checkbox, 'Add' button, 'Strings with suggestions', and 'Add' button. There's also a 'String changed after' section with a date input 'mm/dd/yyyy', a calendar icon, and an 'Add' button. The 'Query examples' section lists several queries with their corresponding Weblate query syntax and an 'Add' button for each:

Query examples	Weblate query syntax	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 area is a 'Search' button.

1.7.1 Basit arama

Any phrase typed into the search box is split into words. Strings containing any of them are shown. To look for an exact phrase, put “the searchphrase” into quotes (both single (') and double (") quotes will work): "this is a quoted string" or 'another quoted string'.

1.7.2 Alanlar

source:METİN

Büyük/küçük harfe duyarsız kaynak dizge arama.

target:METİN

Target string case-insensitive search.

context:METİN

Context string case-insensitive search.

key:METİN

Key string case-insensitive search.

note:METİN

Büyük/küçük harfe duyarsız kaynak dizge açıklaması arama.

location:METİN

Location string case-insensitive search.

priority:SAYI

String priority.

id:NUMBER

Eşsiz dizge belirteci.

added:TARİHSAAT

Timestamp for when the string was added to Weblate.

state:METİN

Search for string states (approved, translated, needs-editing, empty, read-only), supports *Field operators*.

pending:BOOLEAN

String pending for flushing to VCS.

has:METİN

Search for string having attributes - plural, context, suggestion, comment, check, dismissed-check, translation, variant, screenshot, flags, explanation, glossary, note, label.

is:METİN

Search for pending translations (pending). Can also search for all string states (approved, translated, untranslated, needs-editing, read-only).

language:METİN

String target language.

component:METİN

Component slug or name case-insensitive search, see *Bileşen kısaltması* and *Bileşen adı*.

project:METİN

Project slug, see *Adres kısaltması*.

changed_by:METİN

String was changed by author with given username.

changed:DATETIME

String content was changed on date, supports *Field operators*.

change_time:DATETIME

String was changed on date, supports *Field operators*, unlike `changed` this includes event which don't change content and you can apply custom action filtering using `change_action`.

change_action:METİN

Filters on change action, useful together with `change_time`. Accepts English name of the change action, either quoted and with spaces or lowercase and spaces replaced by a hyphen. See *Değişiklik arama* for examples.

check:METİN

String has failing check, see *Denetimler ve düzeltmeler* for check identifiers.

dismissed_check:METİN

String has dismissed check, see *Denetimler ve düzeltmeler* for check identifiers.

comment:METİN

Search in user comments.

resolved_comment:METİN

Çözümlemiş yorumlarda arama.

comment_author:METİN

Filter by comment author.

suggestion:METİN

Search in suggestions.

suggestion_author:METİN

Filter by suggestion author.

explanation:METİN

Açıklamalarda arama.

label:METİN

Etiketlerde arama.

screenshot:METİN

Ekran görüntülerinde arama.

1.7.3 Boolean operators

You can combine lookups using AND, OR, NOT and parentheses to form complex queries. For example: `state:translated AND (source:hello OR source:bar)`

1.7.4 Field operators

You can specify operators, ranges or partial lookups for date or numeric searches:

state:> =translated

State is `translated` or better (approved).

changed:2019

Changed in year 2019.

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

Changed between two given dates.

1.7.5 Exact operators

You can do an exact match query on different string fields using `=` operator. For example, to search for all source strings exactly matching `hello world`, use: `source:="hello world"`. For searching single word expressions, you can skip quotes. For example, to search for all source strings matching `hello`, you can use: `source:=hello`.

1.7.6 Değişiklik arama

4.4 sürümünde geldi.

Searching for history events can be done using `change_action` and `change_time` operators.

For example, searching for strings marked for edit in 2018 can be entered as `change_time:2018 AND change_action:marked-for-edit` or `change_time:2018 AND change_action:"Marked for edit"`.

1.7.7 Kurallı ifadeler

Anywhere text is accepted you can also specify a regular expression as `r"regexp"`.

For example, to search for all source strings which contain any digit between 2 and 5, use `source:r"[2-5]"`.

1.7.8 Predefined queries

You can select out of predefined queries on the search page, this allows you to quickly access the most frequent searches:

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 area features a custom search filter set to '%(count)s word'. A dropdown menu lists various 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 contains a Glossary section and a String information section with fields for Screenshot context, Explanation, Labels, Flags, Source string location, String age, Source string age, and Translation file. The bottom section includes a 'New comment' form and a 'Save' button.

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

1.7.9 Ordering the results

There are many options to order the strings according to your needs:

The screenshot displays the Weblate web interface. At the top, there's a navigation bar with links to Dashboard, Projects, Languages, and Checks. Below this, the breadcrumb trail shows 'WeblateOrg / Django / Czech / Translate'. A progress indicator shows 'translated 96%'. The main content area features a translation form with fields for 'English' and 'Czech'. The English field contains the text 'The string uses three dots (...) instead of an ellipsis character (...)'. The Czech field is empty. Below the form are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A dropdown menu is open, showing options like 'Position and priority', 'Position', 'Priority', 'Labels', 'Source string', 'Target string', 'String age', 'Number of words', 'Number of comments', 'Number of failing checks', 'Key', and 'String location'. The right sidebar contains a 'Glossary' section, a 'String information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file', and a 'New comment' section with a text area and a 'Save' button.

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

1.8 Çeviri iş akışları

Weblate kullanmak, sizi çevirmenlerinize ve kullanıcılarınıza yaklaştıran bir süreçtir. Hangi özellikleri kullanmak istediğinize siz karar verirsiniz.

Aşağıda, farklı Weblate yapılandırmalarının tümü bulunmuyor. Farklı iş akışlarını burada listelenen yaygın uygulamalara göre yapılandırabilirsiniz.

1.8.1 Çeviri erişimi

Erişim denetimi, seçeneklerinin çoğu herhangi bir iş akışına uygulanabileceğinden, iş akışlarında tüm ayrıntılarına girilmemiştir. Çevirilere erişimin nasıl yönetileceği hakkında bilgi almak için ilgili belgelere bakın.

İzleyen bölümlerde, *tüm kullanıcılar*, çeviriye erişimi olan bir kullanıcı anlamına gelir. Proje herkese açıksa, kimliği doğrulanmış herhangi bir kullanıcı veya projede *Çevirme* izni olan bir kullanıcı olabilir.

1.8.2 Çeviri durumları

Çevrilmiş her dizge şu durumlardan birinde olabilir:

Çevrilmemiş

Çeviri yapılmamıştır ve boştur. Dosya biçimine bağlı olarak dosyada bulunabilir ya da bulunmayabilir.

Düzenlenmesi gerekli

Çevirinin düzenlenmesi gereklidir. Genellikle bir kaynak dizge değişikliğinden, belirsiz eşleşmeden veya çevirmen işleminden kaynaklanır. Çeviri, dosya biçimine bağlı olarak dosyada bulunabilir ve düzenlenmesi gerekli şekilde işaretlenebilir (örneğin, Gettext dosyasında `fuzzy` olarak imlenir).

Onaylanmayı bekliyor

Çeviri yapılmış, ancak onaylanmamıştır. Dosyada geçerli bir çeviri olarak bulunur.

Onaylanmış

Çeviri onaylanmıştır. Artık çevirmenler tarafından değiştirilemez. Yalnızca onaylayanlar değiştirebilir. Çevirmenler yalnızca öneri ekleyebilir.

Bu durum yalnızca onaylama özelliği etkinleştirildiğinde kullanılabilir.

Öneriler

Öneriler yalnızca Weblate üzerinde bulunur. Çeviri dosyasında bulunmaz.

Durumlar, çeviri dosyalarında olabildiğince belirtilir.

İpucu: Kullandığınız dosya biçimi durumların kaydedilmesini desteklemiyorsa, değiştirilmemiş dizgeleri düzenlenmesi gerekli olarak işaretlemek için *Değiştirilmemiş çevirileri “Düzenlenmesi gerekli” olarak imleme* eklentisini kullanmak isteyebilirsiniz.

Ayrıca bakınız:

Çeviri türü yetenekleri, Çeviri iş akışları

1.8.3 Doğrudan çeviri

Bu seçenek küçük takımlar için en yaygın kullanılan kurulumdur. Herkes doğrudan çeviri yapabilir. Varsayılan Weblate kurulumudur.

- *Tüm kullanıcılar* çevirileri düzenleyebilir.
- Öneriler, çevirmenlerin çeviride bir değişiklik yapmaktan emin olmadığı durumlarda isteğe bağlı bir değişiklik önerisinde bulunmalarını sağlar.

Ayar	Değer	Not
Onaylama kullanılsın	kapalı	Proje düzeyinde yapılandırılır.
Öneriler kullanılsın	açık	Kullanıcıların emin olmadıkları durumlarda öneri yapabilmeleri açısından yararlıdır.
Öneri oylaması	kapalı	
Öneriler kendiliğinden kabul edilsin	0	
Çevirmenler grubu	<i>Kullanıcılar</i>	Ya da <i>per-project access control</i> ile <i>Çevirmek</i> .
Onaylayıcılar grubu	Kullanılmaz	Kullanılmıyor.

1.8.4 Akran değerlendirmesi

Bu iş akışında, herkes öneri ekleyebilir. Önerilerin onaylanmış olarak kabul edilmesi için başka üyelerin onaylaması gerekir.

- *Tüm kullanıcılar* öneri ekleyebilir.
- *Tüm kullanıcılar* önerilere oy verebilir.
- Önerilerin onaylanmış olarak kabul edilmesi için önceden belirlenmiş sayıda oy alması gerekir.

Ayar	Değer	Not
Onaylama kullanılsın	kapalı	Proje düzeyinde yapılandırılır.
Öneriler kullanılsın	açık	
Öneri oylaması	kapalı	
Öneriler kendiliğinden kabul edilsin	1	Daha fazla akran değerlendirmesi gerekmesi için yüksek bir değer ayarlayabilirsiniz.
Çevirmenler grubu	<i>Kullanıcılar</i>	Ya da <i>per-project access control</i> ile <i>Çevirmek</i> .
Onaylayıcılar grubu	Kullanılmaz	Kullanılmadığında, tüm çevirmenler onaylayabilir.

1.8.5 Özel onaylayanlar

2.18 sürümünde geldi: Weblate 2.18 sürümünden başlayarak normal onaylama iş akışı kullanılabilir.

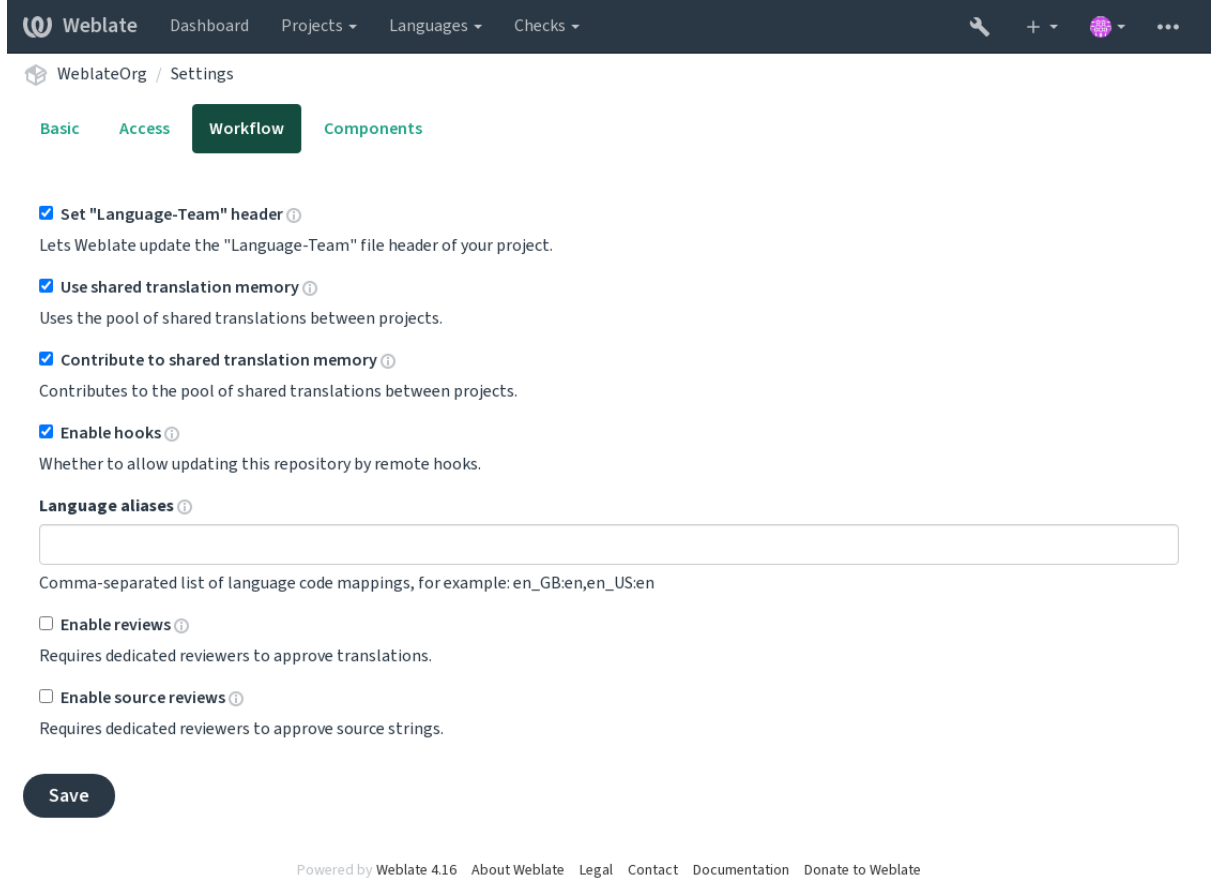
Özel onaylayanlar kullanıldığında, biri çeviri yapan, diğeri de çevirilerin tutarlılığı ve kalitesini sağlamak için bunları gözden geçirip onaylayan iki kullanıcı grubu bulunur.

- *Tüm kullanıcılar* onaylanmamış çevirileri düzenleyebilir.
- *Onaylayan* dizgeleri onaylayabilir ve onayı kaldırabilir.
- *Onaylayan* tüm çevirileri düzenleyebilir (onaylanmışlar ile birlikte).
- Öneriler, onaylanmış dizgelerde değişiklik önermek amacıyla da kullanılabilir.

Ayar	Değer	Not
Onaylama kullanılsın	açık	Proje düzeyinde yapılandırılır.
Öneriler kullanılsın	kapalı	Kullanıcıların emin olmadıkları durumlarda öneri yapabilmeleri açısından yararlıdır.
Öneri oylaması	kapalı	
Öneriler kendiliğinden kabul edilsin	0	
Çevirmenler grubu	<i>Kullanıcılar</i>	Ya da <i>per-project access control</i> ile <i>Çevirmek</i> .
Onaylayıcılar grubu	<i>Onaylayanlar</i>	Ya da <i>per-project access control</i> ile <i>Onayla</i> kullanılabilir.

1.8.6 Onaylama özelliğini açmak

Onaylama, proje yapılandırmasında, proje ayarlarındaki *İş akışı* bölümünden açılabilir (*Yönetim* → *Ayarlar* yoluyla bulabilirsiniz):



The screenshot shows the Weblate web interface. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the breadcrumb 'WeblateOrg / Settings' is visible. The 'Workflow' tab is selected among 'Basic', 'Access', 'Workflow', and 'Components'. The settings list includes:

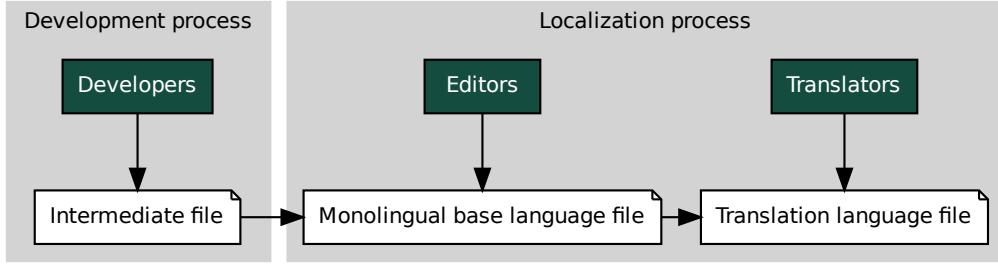
- ☒ **Set "Language-Team" header** ⓘ
Lets Weblate update the "Language-Team" file header of your project.
- ☒ **Use shared translation memory** ⓘ
Uses the pool of shared translations between projects.
- ☒ **Contribute to shared translation memory** ⓘ
Contributes to the pool of shared translations between projects.
- ☒ **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
- ☐ **Enable reviews** ⓘ
Requires dedicated reviewers to approve translations.
- ☐ **Enable source reviews** ⓘ
Requires dedicated reviewers to approve source strings.

A 'Save' button is located at the bottom left of the settings area. The footer of the page reads: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

1.8.7 Kaynak dizgeleri için kalite aracı

Çoğu durumda, özgün kaynak dil dizgeleri geliştiricilerden gelir. Çünkü kodu yazma ve ilk dizgeleri sağlama işini onlar yapar. Bununla birlikte, geliştiricilerin ana dili genellikle kaynak dil değildir ve kaynak dizgeleri için istenilen kaliteyi sağlayamazlar. Ara çeviri bu sorunu çözmenize yardımcı olabilir. Geliştiriciler, çevirmenler ve kullanıcılar arasındaki dizgeleri için kullanılacak bir ek kalite aracı vardır.

Ara dil dosyası ayarlandığında, bu dosya dizgeleri için kaynak olarak kullanılır. Ancak güzelleştirmek için kaynak dile göre düzenlenir. Dizge kaynak dilde hazır olduğunda, çevirmenlerin diğer dillere çevirmesi için de kullanılabilir.

**Ayrıca bakınız:**

Ara dil dosyası, Tek dilli temel dil dosyası, İki dilli ve tek dilli biçimler

1.8.8 Kaynak dizge onayları

Kaynak onaylama kullanılsın seçeneği etkinleştirildiğinde, onaylama işlemi kaynak dizgelere de uygulanabilir. Bu seçenek etkinleştirildikten sonra, kullanıcılar kaynak dizgelerdeki sorunları bildirebilir. Gerçek süreç, iki dilli ya da tek dilli biçimleri kullanıp kullanmadığınıza göre değişir.

Tek dilli biçimlerde, kaynak dizge incelemesi *Özel onaylayanlar* ile benzer şekilde davranır. Kaynak dizge için bir sorun bildirildiğinde dizge *Düzenlenmesi gerekli* olarak işaretlenir.

İki dilli biçimlerde, kaynak dizgelerin doğrudan düzenlenmesine izin verilmez (bunlar genellikle doğrudan kaynak kodundan ayıklanır). Bu durumda çevirmenler tarafından bildirilen dizgelere *Kaynağın incelenmesi gerekli* etiketi eklenir. Bu dizgeleri gözden geçirerek kaynağı düzenlemeniz ya da etiketi kaldırmanız gerekir.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, Özel onaylayanlar, labels, Yorumlar

1.9 Sık sorulan sorular

1.9.1 Yapılandırma

How to create an automated workflow?

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 *Bildirim kancaları* for info on how to do it.
2. Set a push URL at your *Bileşen yapılandırması* in Weblate, this allows Weblate to push changes to your repository.
3. Turn on *İşleme ile itme* on your *Bileşen yapılandırması* in Weblate, this will make Weblate push changes to your repository whenever they happen at Weblate.

Ayrıca bakınız:

Sürekli yerelleştirme, Birleştirme çakışmalarından kaçınmak

How to access repositories over SSH?

Please see *Depolara erişmek* for info on setting up SSH keys.

How to fix merge conflicts in translations?

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).

If you've already encountered a merge conflict, the easiest way to solve all conflicts locally on your machine, is to add Weblate as a remote repository, merge it into upstream and fix any conflicts. Once you push changes back, Weblate will be able to use the merged version without any other special actions.

Not: 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
git merge weblate-one/QA_4_7
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
... # 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
```

Ayrıca bakınız:

How to export the Git repository that Weblate uses?, Sürekli yerelleştirme, Birleştirme çakışmalarından kaçınmak, Weblate istemcisi

How do I translate several branches at once?

Weblate supports pushing translation changes within one *Proje yapılandırması*. For every *Bileşen yapılandırması* which has it turned on (the default behavior), the change made is automatically propagated to others. This way translations are kept synchronized even if the branches themselves have already diverged quite a lot, and it is not possible to simply merge translation changes between them.

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
```

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

How to translate multi-platform projects?

Weblate supports a wide range of file formats (see *Desteklenen dosya biçimleri*) 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 *Bileşenler yapılandırması*) to translate strings for all platforms at once.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

How to export the Git repository that Weblate uses?

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.

What are the options for pushing changes back upstream?

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 *How to create an automated workflow?*).
- 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.

How can I limit Weblate access to only translations, without exposing source code to it?

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.

How can I check whether my Weblate is set up properly?

Weblate includes a set of configuration checks which you can see in the admin interface, just follow the *Performance report* link in the admin interface, or open the `/manage/performance/` URL directly.

Ayrıca bakınız:

Weblate uygulamasını izlemek, Celery durumunu izlemek

Neden tüm işlemler Weblate <noreply@weblate.org> tarafından yapılıyor?

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.

Ayrıca bakınız:

Bileşen yapılandırması

Weblate üzerindeki geçmiş kaybetmeden depodaki dosyalar nasıl taşınır?

To keep the history, comments, or screenshots linked to strings after changing the files location you need to ensure that these strings are never deleted in Weblate. These removals can happen in case the Weblate repository is updated, but the component configuration still points to the old files. This makes Weblate assume that it should delete all the translations.

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

1. Weblate üzerinde etkilenen bileşeni kilitleyin.
2. Bekleyen değişiklikleri işleyin ve depo yukarı akışına birleştirin.
3. Disable receiving webhooks the *Proje yapılandırması*; this prevents Weblate from immediately seeing changes in the repository.
4. Do any needed changes in the repo (for example using `git mv`), push them to the upstream repository.
5. Change the *Bileşen yapılandırması* to match the new setup; upon changing configuration, Weblate will fetch the updated repository and notice the changed locations while keeping existing strings.
6. Bileşenin kilidini açın ve proje yapılandırmasındaki kancaları yeniden etkinleştirin.

1.9.2 Kullanım

How do I review the translations of others?

- There are several review based workflows available in Weblate, see *Çeviri iş akışları*.
- You can subscribe to any changes made in *Bildirimler* and then check others contributions as they come in by e-mail.
- There is a review tool available at the bottom of the translation view, where you can choose to browse translations made by others since a given date.

Ayrıca bakınız:

Çeviri iş akışları

How do I provide feedback on a source string?

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

Ayrıca bakınız:

report-source, [Yorumlar](#)

How can I use existing translations while translating?

- Paylaşılan çeviri belleği ile, Weblate içindeki tüm çeviriler kullanılabilir.
- Var olan çeviri belleği dosyalarını Weblate üzerine aktarabilirsiniz.
- 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.

Ayrıca bakınız:

[Kendiliğinden önerileri yapılandırmak](#), [Kendiliğinden öneriler](#), [Çeviri belleği](#)

Does Weblate update translation files besides translations?

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.

Ayrıca bakınız:

updating-target-files

Where do language definitions come from and how can I add my own?

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.

Ayrıca bakınız:

[Dil tanımları](#)

Can Weblate highlight changes in a fuzzy string?

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.

Why does Weblate still show old translation strings when I've updated the template?

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.

Not: 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 dosyalarını POT ile eşleştirerek güncelleme (msgmerge)*.

Ayrıca bakınız:

updating-target-files

1.9.3 Sorun çözme

Requests sometimes fail with “too many open files” error

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
```

Ayrıca bakınız:

DATA_DIR

When accessing the site I get a “Bad Request (400)” error

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"]
```

Ayrıca bakınız:

Allowed hosts kurulumu

What does mean “There are more files for the single language (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.

- Kaynak dile çeviri isteniyorsa, lütfen bileşen ayarlarındaki *Kaynak dil* seçeneğini değiştirin. Kaynak dili *İngilizce (Geliştirici)* olarak seçmek ya da *Kaynak dizgeleri için kalite aracı* kullanmak isteyebilirsiniz.
- Kaynak dil için çeviri dosyasına gerek yoksa, lütfen depodan kaldırın.
- Kaynak dilin çeviri dosyasına gerek duyulduğu, ancak Weblate tarafından yok sayılmasının gerektiği durumlarda, katılmaması için *Dil süzgeci* seçeneğini ayarlayın.

İpucu: 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 *Dil kodlarını işlemek* for more details.

1.9.4 Özellikler

Does Weblate support other VCSes than Git and Mercurial?

Weblate currently does not have native support for anything other than *Git* (with extended support for *GitHub çekme istekleri*, *Gerrit* and *Subversion*) and *Mercurial*, but it is possible to write backends for other VCSes.

You can also use *Uzak Git yardımcıları* in Git to access other VCSes.

Weblate also supports VCS-less operation, see *Yerel dosyalar*.

Not: 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.

Ayrıca bakınız:

Sürüm denetimi bütünleştirmesi

How does Weblate credit translators?

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.

Ayrıca bakınız:

list_translators, *../devel/reporting*

Why does Weblate force showing all PO files in a single tree?

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.

4.2 sürümünde değişti: Çevirmenler, bir projenin tüm bileşenlerini bir bütün olarak belirli bir dile çevirebilir.

Why does Weblate use language codes such sr_Latn or zh_Hant?

These are language codes defined by [RFC 5646](#) to better indicate that they are really different languages instead previously wrongly used modifiers (for @latin variants) or country codes (for Chinese).

Weblate still understands legacy language codes and will map them to current one - for example sr@latin will be handled as sr_Latn or zh@CN as zh_Hans.

Not: Weblate defaults to POSIX style language codes with underscore, see [Dil tanımları](#) for more details.

Ayrıca bakınız:

[Dil tanımları](#), [Dil kodu biçemi](#), [Yeni çevirileri eklemek](#)

1.10 Desteklenen dosya biçimleri

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.

Ayrıca bakınız:

[Translation Related File Formats](#)

Not: When choosing a file format for your application, it's better to stick some well established format in the tool-kit/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 İki dilli ve tek dilli biçimler

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 dizgeleri](#)). 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 resources](#)). 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 [Tek dilli temel dil dosyası](#) within Weblate, though the naming might vary in your paradigm.

Additionally this workflow can be extended by utilizing [Ara dil dosyası](#) to include strings provided by developers, but not to be used as is in the final strings.

1.10.2 Kendiliğinden algılama

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 Çeviri türü yetenekleri

Desteklenen tüm biçimlerdeki yetenekler:

Biçim	Dil bilgi- si ^{Sayfa 78, 1}	Çoğul- lar ^{Sayfa 78, 2}	Açıklama- lar ^{Sayfa 78, 3}	Bağ- lam ^{Sayfa 78, 4}	Ko- num ^{Sayfa 78, 5}	İm- ler ^{Sayfa 78, 8}	Ek durum- lar ^{Sayfa 78, 6}
<i>GNU gettext</i>	iki dilde	evet	evet	evet	evet	evet ⁹	düzen- lenmesi gerekli
<i>Tek dilli gettext</i>	tek	evet	evet	evet	evet	evet ^{Sayfa 78, 9}	düzen- lenmesi gerekli
<i>XLIFF</i>	ikisi de	evet	evet	evet	evet	evet ¹⁰	düzen- lenmesi gerekli, onaylanmış
<i>Java özellikleri</i>	ikisi de	hayır	evet	hayır	hayır	hayır	
<i>mi18n lang dosyaları</i>	tek	hayır	evet	hayır	hayır	hayır	
<i>GWT özel- likleri</i>	tek	evet	evet	hayır	hayır	hayır	
<i>Joomla! çe- virileri</i>	tek	hayır	evet	hayır	evet	hayır	
<i>Qt Linguist .ts</i>	ikisi de	evet	evet	hayır	evet	evet ^{Sayfa 78, 10}	düzen- lenmesi gerekli
<i>Android st- ring resour- ces</i>	tek	evet	evet ⁷	hayır	hayır	evet ^{Sayfa 78, 10}	
<i>Apple iOS dizgeleri</i>	ikisi de	hayır	evet	hayır	hayır	hayır	
<i>PHP dizge- leri</i>	tek	hayır ¹¹	evet	hayır	hayır	hayır	
<i>JSON dos- yaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>JSON i18next dosyaları</i>	tek	evet	hayır	hayır	hayır	hayır	
<i>go-i18n JSON dosyaları</i>	tek	evet	hayır	hayır	hayır	hayır	
<i>gotext JSON dosyaları</i>	tek	evet	evet	hayır	evet	hayır	
<i>ARB dosyası</i>	tek	evet	evet	hayır	hayır	hayır	
<i>WebEx- tension JSON</i>	tek	evet	evet	hayır	hayır	hayır	

sonraki sayfaya devam

Tablo 1 – önceki sayfadan devam

Bişim	Dil bilgi- si ^{Sayfa 78, 1}	Çoğul- lar ^{Sayfa 78, 2}	Açıklama- lar ^{Sayfa 78, 3}	Bağ- lam ^{Sayfa 78, 4}	Ko- num ^{Sayfa 78, 5}	İm- ler ^{Sayfa 78, 8}	Ek durum- lar ^{Sayfa 78, 6}
<i>.XML kaynak dosyaları</i>	tek	hayır	evet	hayır	hayır	evet ^{Sayfa 78, 10}	
<i>Resource-Dictionary dosyaları</i>	tek	hayır	hayır	hayır	hayır	evet ^{Sayfa 78, 10}	
<i>CSV dosyaları</i>	ikisi de	hayır	evet	evet	evet	hayır	düzen- lenmesi gerekli
<i>YAML dosyaları</i>	tek	hayır	evet	hayır	hayır	hayır	
<i>Ruby YAML dosyaları</i>	tek	evet	evet	hayır	hayır	hayır	
<i>DTD dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Düz XML dosyaları</i>	tek	hayır	hayır	hayır	hayır	evet ^{Sayfa 78, 10}	
<i>Windows RC dosyaları</i>	tek	hayır	evet	hayır	hayır	hayır	
<i>Excel Open XML</i>	tek	hayır	evet	evet	evet	hayır	düzen- lenmesi gerekli
<i>Uygulama mağazası üst veri dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Alt yazı dosyaları</i>	tek	hayır	hayır	hayır	evet	hayır	
<i>HTML dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>OpenDocument biçimi</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>IDML biçimi</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>INI çevirileri</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Inno Setup INI çevirileri</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>TermBase eXchange biçimi</i>	iki dilde	hayır	evet	hayır	hayır	evet ^{Sayfa 78, 10}	
<i>Metin dosyaları</i>	tek	hayır	hayır	hayır	hayır	hayır	
<i>Stringsdict biçimi</i>	tek	evet	evet	hayır	hayır	hayır	
<i>Fluent biçimi</i>	tek	hayır ¹²	evet	hayır	hayır	hayır	

Salt okunur dizgeler

3.10 sürümünde geldi.

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 resources*), but can be emulated in others by adding a `read-only` flag, see *Parametreleri kullanarak davranışı özelleştirmek*.

1.10.4 GNU gettext

Özgür yazılımları çevirmek için en yaygın olarak kullanılan biçim.

Dosyada depolanan bağlamsal bilgiler, üst bilgileri ayarlanarak veya ilgili kaynak dosyalarla bağlantı kurularak sağlanabilir.

İki dilli gettext PO dosyası genellikle şöyle görünür:

```
#: 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"
msgid "None"
msgstr "Žádný"
```

Tipik Weblate *Bileşen yapılandırması*

Dosya maskesi	po/* .po
Tek dilli temel dil dosyası	Boş
Yeni çeviriler kalıbı	po/messages .pot
Dosya biçimi	Gettext PO dosyası

Ayrıca bakınız:

devel/gettext, devel/sphinx, Vikipedi üzerinde Gettext, PO Files, “*configure*” dosyasındaki *ALL_LINGUAS* değişkeni güncelleme, *Gettext çıktısını özelleştirme*, *LINGUAS* dosyasını güncelleme, *MO* dosyaları üretme, *PO* dosyalarını *POT* ile eşleştirerek güncelleme (*msgmerge*)

¹ Bilgi almak için: *İki dilli ve tek dilli biçimler*

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 Bilgi almak için: *Parametreleri kullanarak davranışı özelleştirmek*

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 Plurals are handled in the syntax of the strings and not exposed as plurals in Weblate.

Tek dilli 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.

Tek dilli gettext PO dosyası genellikle şöyle görünür:

```
#: 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"
```

Temel dil dosyası şöyle olacaktır:

```
#: 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"
```

Tipik Weblate *Bileşen yapılandırması*

Dosya maskesi	po/* .po
Tek dilli temel dil dosyası	po/en .po
Yeni çeviriler kalıbı	po/messages .pot
Dosya biçimi	<i>Gettext PO dosyası (tek dilli)</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, birkaç çeşitte XLIFF desteği sunar:

XLIFF çeviri dosyası

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

XLIFF with placeables support

Standard XLIFF supporting placeables and other XML elements.

gettext uzantılı XLIFF

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

Ayrıca bakınız:

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

Çeviri durumları

3.3 sürümünde değişti: 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*.

Ayrıca bakınız:

Özel onaylayanlar

Whitespace and newlines 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.

Örnek:

```
<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 *Parametreleri kullanarak davranışı özelleştirmek*) 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>
```

The `font` attribute is parsed for font family, size and weight, the above example shows all of that, though only font family is required. Any whitespace in the font family is converted to underscore, so `Source Sans Pro` becomes `Source_Sans_Pro`, please keep that in mind when naming the font group (see *Yazı tiplerini yönetmek*).

Dizge anahtarları

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>Bileşen yapılandırması</i> for bilingual XLIFF	
Dosya maskesi	<code>localizations/*.xliff</code>
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	<code>localizations/en-US.xliff</code>
Dosya biçimi	<i>XLIFF çeviri dosyası</i>

Typical Weblate <i>Bileşen yapılandırması</i> for monolingual XLIFF	
Dosya maskesi	<code>localizations/*.xliff</code>
Tek dilli temel dil dosyası	<code>localizations/en-US.xliff</code>
Yeni çeviriler kalıbı	<code>localizations/en-US.xliff</code>
Dosya biçimi	<i>XLIFF çeviri dosyası</i>

Ayrıca bakınız:

[Vikipedi üzerinde XLIFF](#), [XLIFF](#), [font attribute in XLIFF 1.2](#), [maxwidth attribute in XLIFF 1.2](#)

1.10.6 Java özellikleri

Çeviriler için doğal Java biçimi.

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.

Not: Loading escape sequences works in UTF-8 mode as well, so please be careful choosing the correct encoding set to match your application needs.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	<code>src/app/Bundle_*.properties</code>
Tek dilli temel dil dosyası	<code>src/app/Bundle.properties</code>
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Java Properties (ISO-8859-1)</i>

Ayrıca bakınız:

[Vikipedi üzerinde Java özellikleri](#), [Mozilla and Java properties files](#), [mi18n lang dosyaları](#), [GWT özellikleri](#), [updating-target-files](#), [Java özellikleri dosyasını biçimlendirme](#), [Çeviri dosyalarını temizleme](#)

1.10.7 mi18n lang dosyaları

4.7 sürümünde geldi.

File format used for JavaScript localization by `mi18n`. Syntactically it matches *Java özellikleri*.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	*.lang
Tek dilli temel dil dosyası	en-US.lang
Yeni çeviriler kalıbı	Boş
Dosya biçimi	mi18n lang dosyası

Ayrıca bakınız:

[mi18n](#), [Mozilla and Java properties files](#), [Java özellikleri](#), [updating-target-files](#), [Java özellikleri dosyasını biçimlendirme](#), [Çeviri dosyalarını temizleme](#)

1.10.8 GWT özellikleri

Native GWT format for translations.

GWT properties are usually used as monolingual translations.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	src/app/Bundle_*.properties
Tek dilli temel dil dosyası	src/app/Bundle.properties
Yeni çeviriler kalıbı	Boş
Dosya biçimi	GWT özellikleri

Ayrıca bakınız:

[GWT localization guide](#), [GWT Internationalization Tutorial](#), [Mozilla and Java properties files](#), [updating-target-files](#), [Java özellikleri dosyasını biçimlendirme](#), [Çeviri dosyalarını temizleme](#)

1.10.9 INI çevirileri

4.1 sürümünde geldi.

Çeviriler için INI dosya biçimi.

INI translations are usually used as monolingual translations.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	language/*.ini
Tek dilli temel dil dosyası	language/en.ini
Yeni çeviriler kalıbı	Boş
Dosya biçimi	INI dosyası

Not: Weblate only extracts keys from sections within an INI file. In case your INI file lacks sections, you might want to use *Joomla! çevirileri* or *Java özellikleri* instead.

Ayrıca bakınız:

[INI Files](#), [Java özellikleri](#), [Joomla! çevirileri](#), [Inno Setup INI çevirileri](#)

1.10.10 Inno Setup INI çevirileri

4.1 sürümünde geldi.

Çeviriler için Inno Setup INI dosya biçimi.

Inno Setup INI çevirileri genellikle tek dilli çeviriler olarak kullanılır.

Not: The only notable difference to *INI çevirileri* is in supporting %n and %t placeholders for line break and tab.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	language/*.isl
Tek dilli temel dil dosyası	language/en.isl
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Inno Setup INI dosyası</i>

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

Ayrıca bakınız:

[INI Files](#), [Joomla! çevirileri](#), [INI çevirileri](#)

1.10.11 Joomla! çevirileri

2.12 sürümünde geldi.

Çeviriler için doğal Joomla! biçimi.

Joomla translations are usually used as monolingual translations.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	language/*/com_foobar.ini
Tek dilli temel dil dosyası	language/en-GB/com_foobar.ini
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Joomla dil dosyası</i>

Ayrıca bakınız:

[Mozilla and Java properties files](#), [INI çevirileri](#), [Inno Setup INI çevirileri](#)

1.10.12 Qt Linguist .ts

Translation format used in Qt based applications.

Qt Linguist files are used as both bilingual and monolingual translations.

Typical Weblate <i>Bileşen yapılandırması</i> when using as bilingual	
Dosya maskesi	i18n/app.*.ts
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	i18n/app.de.ts
Dosya biçimi	<i>Qt Linguist Translation File</i>

Typical Weblate <i>Bileşen yapılandırması</i> when using as monolingual	
Dosya maskesi	i18n/app.*.ts
Tek dilli temel dil dosyası	i18n/app.en.ts
Yeni çeviriler kalıbı	i18n/app.en.ts
Dosya biçimi	Qt Linguist Translation File

Ayrıca bakınız:

Qt Linguist manual, Qt .ts, *İki dilli ve tek dilli biçimler*

1.10.13 Android string resources

Android specific file format for translating applications.

Android string resources are monolingual, the *Tek dilli temel dil dosyası* is stored in a different location from the other files – res/values/strings.xml.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	res/values-*/strings.xml
Tek dilli temel dil dosyası	res/values/strings.xml
Yeni çeviriler kalıbı	Boş
Dosya biçimi	Android String Resource

Ayrıca bakınız:

Android string resources documentation, Android string resources

Not: 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>

İpucu: 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 dizgeleri

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

Apple iOS dizgeleri genellikle tek dilli çeviriler olarak kullanılır.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Resources/*.lproj/Localizable.strings
Tek dilli temel dil dosyası	Resources/en.lproj/Localizable.strings or Resources/Base.lproj/Localizable.strings
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>iOS dizgeleri (UTF-8)</i>

Ayrıca bakınız:

Stringsdict biçimi, Apple “strings files” documentation, Message Catalog File Format in PWG 5100.13, Mac OSX strings

1.10.15 PHP dizgeleri

PHP translations are usually monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Örnek dosya:

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

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	lang/*/texts.php
Tek dilli temel dil dosyası	lang/en/texts.php
Yeni çeviriler kalıbı	lang/en/texts.php
Dosya biçimi	<i>PHP dizgeleri</i>

Laravel PHP dizgeleri

4.1 sürümünde değişti.

The Laravel PHP localization files are supported as well with plurals:

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

Ayrıca bakınız:

PHP, Laravel ile yerelleştirme

1.10.16 JSON dosyaları

2.0 sürümünde geldi.

2.16 sürümünde değişti: Since Weblate 2.16 and with [translate-toolkit](#) at-least 2.2.4, nested structure JSON files are supported as well.

4.3 sürümünde değişti: 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 currently supports several variants of JSON translations:

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

JSON translations are usually monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Örnek dosya:

```
{
  "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": ""
  }
}
```

İpucu: 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"
  }
}
```

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	langs/translation-*.json
Tek dilli temel dil dosyası	langs/translation-en.json
Yeni çeviriler kalıbı	Boş
Dosya biçimi	'JSON iç içe geçmiş yapı dosyası'

Ayrıca bakınız:

JSON, updating-target-files, *JSON çıktısını özelleştirme*, *Çeviri dosyalarını temizleme*,

1.10.17 JSON i18next dosyaları

2.17 sürümünde değişti: Since Weblate 2.17 and with [translate-toolkit](#) at-least 2.2.5, i18next JSON files with plurals are supported as well.

4.15.1 sürümünde değişti: Support for v4 variant of this format was added.

İpucu: 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 translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Not: Weblate supports the i18next JSON v3 and v4 variants. Please choose correct file format matching your environment.

The v2 and v1 variants are mostly compatible with v3, with exception of how plurals are handled.

Örnek dosya:

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

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	langs/*.json
Tek dilli temel dil dosyası	langs/en.json
Yeni çeviriler kalıbı	Boş
Dosya biçimi	i18next JSON dosyası v3

Ayrıca bakınız:

JSON, i18next JSON biçimi, updating-target-files, *JSON çıktısını özelleştirme*, *Çeviri dosyalarını temizleme*

1.10.18 go-i18n JSON dosyaları

4.1 sürümünde geldi.

go-i18n translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

4.16 sürümünde değişti: Weblate supports both v1 and v2 formats, please choose corresponding version when selecting file format.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	langs/*.json
Tek dilli temel dil dosyası	langs/en.json
Yeni çeviriler kalıbı	Boş
Dosya biçimi	go-i18n v1 JSON dosyası

Ayrıca bakınız:

[JSON](#), [go-i18n](#), [updating-target-files](#), [JSON çıktısını özelleştirme](#), [Çeviri dosyalarını temizleme](#),

1.10.19 gotext JSON dosyaları

4.15.1 sürümünde geldi.

gotext translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	internal/translations/locales/*/messages.gotext.json
Tek dilli temel dil dosyası	internal/translations/locales/en-GB/messages.gotext.json
Yeni çeviriler kalıbı	Boş
Dosya biçimi	gotext JSON dosyası

Ayrıca bakınız:

[JSON](#), [I18n in Go: Managing Translations](#), [updating-target-files](#), [JSON çıktısını özelleştirme](#), [Çeviri dosyalarını temizleme](#),

1.10.20 ARB dosyası

4.1 sürümünde geldi.

ARB translations are monolingual, so it is recommended to specify a base file with (what is most often the) English strings.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	lib/l10n/intl_*.arb
Tek dilli temel dil dosyası	lib/l10n/intl_en.arb
Yeni çeviriler kalıbı	Boş
Dosya biçimi	ARB dosyası

Ayrıca bakınız:

[JSON](#), [Application Resource Bundle Specification](#), [Internationalizing Flutter apps](#), [updating-target-files](#), [JSON çıktısını özelleştirme](#), [Çeviri dosyalarını temizleme](#)

1.10.21 WebExtension JSON

2.16 sürümünde geldi: This is supported since Weblate 2.16 and with [translate-toolkit](#) at-least 2.2.4.

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

Not: 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.

Örnek dosya:

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

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	<code>_locales/*/messages.json</code>
Tek dilli temel dil dosyası	<code>_locales/en/messages.json</code>
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>WebExtension JSON file</i>

Ayrıca bakınız:

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

1.10.22 .XML kaynak dosyaları

2.3 sürümünde geldi.

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.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Resources/Language.*.resx
Tek dilli temel dil dosyası	Resources/Language.resx
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>.NET kaynak dosyası</i>

Ayrıca bakınız:

[.NET Resource files \(.resx\)](#), [updating-target-files](#), [Çeviri dosyalarını temizleme](#)

1.10.23 ResourceDictionary dosyaları

4.13 sürümünde geldi.

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

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Languages/*.xaml
Tek dilli temel dil dosyası	Language/en.xaml
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>ResourceDictionary dosyası</i>

Ayrıca bakınız:

[Flat XML](#), [Düz XML dosyaları](#), [updating-target-files](#), [Çeviri dosyalarını temizleme](#)

1.10.24 CSV dosyaları

2.4 sürümünde geldi.

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 *Tek dilli temel dil dosyası* when your files are monolingual (see [İki dilli ve tek dilli biçimler](#)).

İpucu: Varsayılan olarak, CSV biçimi dosya kodlamasının kendiliğinden algılanmasını sağlar. Bu davranış, bazı uç durumlarda güvenilir olmaz ve başarımın düşmesine yol açabilir. Bunu önlemek için lütfen kodlamalı dosya biçimi çeşidini seçin (*CSV dosyası (UTF-8)* gibi).

Uyarı: The CSV format currently automatically detects the dialect of the CSV file. In some cases the automatic detection might fail and you will get mixed results. This is especially true for CSV files with newlines in the values. As a workaround it is recommended to omit quoting characters.

Örnek dosya:

Thank you for using Weblate.,Děkuje me za použití Weblate.

Typical Weblate <i>Bileşen yapılandırması</i> for bilingual CSV	
Dosya maskesi	locale/*.csv
Tek dilli temel dil dosyası	<i>Boş</i>
Yeni çeviriler kalıbı	locale/en.csv
Dosya biçimi	<i>CSV dosyası</i>

Typical Weblate <i>Bileşen yapılandırması</i> for monolingual CSV	
Dosya maskesi	locale/*.csv
Tek dilli temel dil dosyası	locale/en.csv
Yeni çeviriler kalıbı	locale/en.csv
Dosya biçimi	<i>Basit CSV dosyası</i>

Çok değerli CSV dosyası

4.13 sürümünde geldi.

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

Ayrıca bakınız:

CSV

1.10.25 YAML dosyaları

2.9 sürümünde geldi.

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

YAML dosyası örneği:

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

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	translations/messages/*.yaml
Tek dilli temel dil dosyası	translations/messages.en.yaml
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>YAML dosyası</i>

Ayrıca bakınız:

YAML, *Ruby YAML dosyaları*

1.10.26 Ruby YAML dosyaları

2.9 sürümünde geldi.

Ruby i18n YAML files with language as root node.

Ruby i18n YAML dosyası örneği:

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

Tipik Weblate *Bileşen yapılandırması*

Dosya maskesi	translations/messages.*.yaml
Tek dilli temel dil dosyası	translations/messages.en.yaml
Yeni çeviriler kalıbı	Boş
Dosya biçimi	Ruby YAML dosyası

Ayrıca bakınız:

[YAML](#), [YAML dosyaları](#)

1.10.27 DTD dosyaları

2.18 sürümünde geldi.

Örnek DTD dosyası:

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

Tipik Weblate *Bileşen yapılandırması*

Dosya maskesi	locale/*.dtd
Tek dilli temel dil dosyası	locale/en.dtd
Yeni çeviriler kalıbı	Boş
Dosya biçimi	DTD dosyası

Ayrıca bakınız:

[Mozilla DTD format](#)

1.10.28 Düz XML dosyaları

3.9 sürümünde geldi.

Düz XML dosyası örneği:

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

Tipik Weblate *Bileşen yapılandırması*

Dosya maskesi	locale/*.xml
Tek dilli temel dil dosyası	locale/en.xml
Yeni çeviriler kalıbı	Boş
Dosya biçimi	Düz XML dosyası

Ayrıca bakınız:

[Flat XML](#)

1.10.29 Windows RC dosyaları

4.1 sürümünde değişti: Windows RC dosyaları desteği yeniden yazıldı.

Not: Bu biçimin desteği şu anda beta sürümünde. Denemelerin geri bildirimini bekliyoruz.

Windows RC dosyası örneği:

```
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
```

Tipik Weblate *Bileşen yapılandırması*

Dosya maskesi	lang/*.rc
Tek dilli temel dil dosyası	lang/en-US.rc
Yeni çeviriler kalıbı	lang/en-US.rc
Dosya biçimi	RC dosyası

Ayrıca bakınız:

[Windows RC files](#)

1.10.30 Uygulama mağazası üst veri dosyaları

3.5 sürümünde geldi.

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

Üst veriler, Weblate tarafından çevrilmek üzere ayrı dizgeler olarak sunulacak birkaç metin dosyasından oluşur.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	fastlane/android/metadata/*
Tek dilli temel dil dosyası	fastlane/android/metadata/en-US
Yeni çeviriler kalıbı	fastlane/android/metadata/en-US
Dosya biçimi	<i>App store üst veri dosyaları</i>

İpucu: In case you don't want to translate certain strings (for example changelogs), mark them read-only (see [Parametreleri kullanarak davranışı özelleştirmek](#)). This can be automated by the [Toplu düzenleme](#).

1.10.31 Alt yazı dosyaları

3.7 sürümünde geldi.

Weblate çeşitli alt yazı dosyalarını çevirebilir:

- SubRip alt yazı dosyası (*.srt)
- MicroDVD alt yazı dosyası (*.sub)
- Advanced Substation Alpha alt yazı dosyası (*.ass)
- Substation Alpha alt yazı dosyası (*.ssa)

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	path/*.srt
Tek dilli temel dil dosyası	path/en.srt
Yeni çeviriler kalıbı	path/en.srt
Dosya biçimi	<i>SubRip alt yazı dosyası</i>

Ayrıca bakınız:

[Subtitles](#)

1.10.32 Excel Open XML

3.2 sürümünde geldi.

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 dosyaları

4.1 sürümünde geldi.

Not: Bu biçimin desteği şu anda beta sürümünde. Denemelerin geri bildirimini bekliyoruz.

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

Ayrıca bakınız:

[HTML](#)

1.10.34 Metin dosyaları

4.6 sürümünde geldi.

Not: Bu biçimin desteği şu anda beta sürümünde. Denemelerin geri bildirimini bekliyoruz.

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

Bu biçimin üç çeşidi vardır:

- Düz metin dosyası
- DokuWiki metin dosyası
- MediaWiki metin dosyası

Ayrıca bakınız:

[Simple Text Documents](#)

1.10.35 OpenDocument biçimi

4.1 sürümünde geldi.

Not: Bu biçimin desteği şu anda beta sürümünde. Denemelerin geri bildirimini bekliyoruz.

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

Ayrıca bakınız:

[OpenDocument Format](#)

1.10.36 IDML biçimi

4.1 sürümünde geldi.

Not: Bu biçimin desteği şu anda beta sürümünde. Denemelerin geri bildirimini bekliyoruz.

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

1.10.37 TermBase eXchange biçimi

4.5 sürümünde geldi.

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

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	tbx/* .tbx
Tek dilli temel dil dosyası	Boş
Yeni çeviriler kalıbı	Boş
Dosya biçimi	TermBase eXchange dosyası

Ayrıca bakınız:

[TBX on Wikipedia](#), [TBX](#), [Sözlük](#)

1.10.38 Stringsdict biçimi

4.8 sürümünde geldi.

Not: Bu biçimin desteği şu anda beta sürümünde. Denemelerin geri bildirimini bekliyoruz.

XML based format used by Apple which is able to store plural forms of a string.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	Resources/* .lproj/Localizable.stringsdict
Tek dilli temel dil dosyası	Resources/en.lproj/Localizable.stringsdict ya da Resources/Base.lproj/Localizable.stringsdict
Yeni çeviriler kalıbı	Boş
Dosya biçimi	Stringsdict dosyası

Ayrıca bakınız:

[Apple iOS dizgeleri](#), [Stringsdict dosya biçimi](#)

1.10.39 Fluent biçimi

4.8 sürümünde geldi.

Not: Bu biçimin desteği şu anda beta sürümünde. Denemelerin geri bildirimini bekliyoruz.

Fluent is a monolingual text format that focuses on asymmetric localization: a simple string in one language can map to a complex multi-variant translation in another language.

Tipik Weblate <i>Bileşen yapılandırması</i>	
Dosya maskesi	locales/*/messages.ftl
Tek dilli temel dil dosyası	locales/en/messages.ftl
Yeni çeviriler kalıbı	<i>Boş</i>
Dosya biçimi	<i>Fluent dosyası</i>

Ayrıca bakınız:

[Project Fluent website](#)

1.10.40 Diğer biçimlerin desteği

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](#).

Ayrıca bakınız:

[Translation Related File Formats](#)

1.11 Sürüm denetimi bütünleştirmesi

Weblate currently supports *Git* (with extended support for *GitHub çekme istekleri*, *GitLab birleştirme istekleri*, *Gitea çekme isteği*, *Gerrit*, *Subversion* and *Bitbucket sunucusu çekme isteği*) and *Mercurial* as version control back-ends.

1.11.1 Depolara erişmek

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.

Hosted Weblate üzerinden depolara erişmek

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 *Kaynak kod deposu* and *Depo itme adresi* using the SSH protocol (for example [git@github.com:WeblateOrg/weblate.git](ssh://git@github.com:WeblateOrg/weblate.git)).

SSH depoları

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

Uyarı: On GitHub, each key can only be used once, see [GitHub depoları](#) and [Hosted Weblate üzerinden depolara erişmek](#).

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

In case adjustment is needed, do so from the Weblate admin interface:

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
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWDxygRSjmBGwjV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhlLwp1uYlW
Weblate
```

[Download private key](#)

Known host keys

Hostname	Key type	Fingerprint
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-ed25519	+DiY3wvV6TuJJhbpZisF/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](#)

Weblate SSH anahtarı

The Weblate public key is visible to all users browsing the *About* page.

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

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

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

SSH sunucu anahtarlarını doğrulamak

Weblate automatically stores the SSH host keys on first access and remembers them for further use.

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:

Weblate
 Dashboard Projects Languages Checks

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 nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8 (ssh-rsa), please verify that it is correct.

Added host key for github.com with fingerprint +DiY3wvV6TuJJhbpZisF/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
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWDxygRSjmBGwJv0ZMgkux4GAuPY69M6ZeWbC1skyQJxPcgyFCvoZniU1yVhLwlp1uYlW
Weblate
```

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-ed25519	+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

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 depoları

Access via SSH is possible (see [SSH depoları](#)), but in case you need to access more than one repository, you will hit a GitHub limitation on allowed SSH key usage (since each key can be used only once).

In case the *İtme işleminin yapılacağı dal* 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 anahtarı](#)) 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.

Ayrıca bakınız:

[Hosted Weblate üzerinden depolara erişmek](#)

İç Weblate adresleri

Share one repository setup between different components by referring to its placement as `weblate://project/component` in other(linked) components. This way linked components use the VCS repository configuration of the main(referenced) component.

Uyarı: Removing main component also removes linked components.

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.

Bunu kullanma nedenleri:

- 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](#)).
- Some add-ons can operate on multiple components sharing one repository, for example [Git işlemelerini bir araya toplar](#).

HTTPS depoları

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`.

Not: If your username or password contains special characters, those have to be URL encoded, for example `https://user%40example.com:%24password%23@bitbucket.org/...`

Vekil sunucu kullanmak

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
```

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

Ayrıca bakınız:

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

1.11.2 Git

İpucu: Weblate needs Git 2.12 or newer.

Ayrıca bakınız:

See [Depolara erişmek](#) for info on how to access different kinds of repositories.

Git (itme dayatması ile)

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.

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

Git yapılandırmasını özelleştirmek

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`.

Uzak Git yardımcıları

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
```

Uyarı: 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 çekme istekleri

2.3 sürümünde geldi.

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](#) pushes changes directly to a repository, while [GitHub çekme istekleri](#) creates pull requests. The latter is not needed for merely accessing Git repositories.

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 [Sürüm denetimi sistemi](#).

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri görmek, `GITHUB_CREDENTIALS`

1.11.4 GitLab birleştirme istekleri

3.9 sürümünde geldi.

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.

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 *GitLab birleştirme istekleri* creates merge request.

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 *Sürüm denetimi sistemi*.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri görmek, `GITLAB_CREDENTIALS`

1.11.5 Gitea çekme isteği

4.12 sürümünde geldi.

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.

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 *Gitea çekme isteği* creates pull requests.

You need to configure API credentials (*GITEA_CREDENTIALS*) in the Weblate settings to make this work. Once configured, you will see a *Gitea* option when selecting *Sürüm denetimi sistemi*.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri görmek, `GITEA_CREDENTIALS`

1.11.6 Bitbucket sunucusu çekme isteği

4.16 sürümünde geldi.

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.

Uyarı: 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 sunucusu çekme isteği* 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 *Sürüm denetimi sistemi*.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri görmek, `BITBUCKETSERVER_CREDENTIALS`

1.11.7 Pagure birleştirme istekleri

4.3.2 sürümünde geldi.

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.

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 [Pagure birleştirme istekleri](#) creates merge request.

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 [Sürüm denetimi sistemi](#).

Ayrıca bakınız:

[Weblate üzerindeki değişiklikleri itmek](#), `PAGURE_CREDENTIALS`

1.11.8 Gerrit

2.2 sürümünde geldi.

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.

The Gerrit documentation has the details on the configuration necessary to set up such repositories.

1.11.9 Mercurial

2.1 sürümünde geldi.

Mercurial is another VCS you can use directly in Weblate.

Not: It should work with any Mercurial version, but there are sometimes incompatible changes to the command-line interface which breaks Weblate integration.

Ayrıca bakınız:

See [Depolara erişmek](#) for info on how to access different kinds of repositories.

1.11.10 Subversion

2.8 sürümünde geldi.

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.

Not: 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.

2.19 sürümünde değişti: Before this, only repositories using the standard layout were supported.

Subversion kimlik doğrulama bilgileri

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
```

Ayrıca bakınız:

`DATA_DIR`

1.11.11 Yerel dosyalar

1.11.12 Git

İpucu: Underneath, this uses [Git](#). It requires Git installed and allows you to switch to using Git natively with full history of your translations.

3.8 sürümünde geldi.

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 Weblate REST API uygulaması

2.6 sürümünde geldi: REST API, Weblate 2.6 sürümünden başlayarak kullanılabilir.

The API is accessible on the `/api/` URL and it is based on [Django REST framework](#). You can use it directly or by [Weblate istemcisi](#).

1.12.1 Authentication and generic parameters

The public project API is available without authentication, though unauthenticated requests are heavily throttled (by default to 100 requests per day), so it is recommended to use authentication. The authentication uses a token, which you can get in your profile. Use it in the `Authorization` header:

ANY /

Generic request behaviour for the API, the headers, status codes and parameters here apply to all endpoints as well.

Sorgu parametreleri

- **format** – Response format (overrides [Accept](#)). Possible values depends on REST framework setup, by default `json` and `api` are supported. The latter provides web browser interface for API.
- **page** – Returns given page of paginated results (use `next` and `previous` fields in response to automate the navigation).

İstek üst bilgileri

- [Accept](#) – the response content type depends on [Accept](#) header

- **Authorization** – optional token to authenticate as **Authorization: Token YOUR-TOKEN**

Yanıt üst bilgileri

- **Content-Type** – this depends on **Accept** header of request
- **Allow** – list of allowed HTTP methods on object

Yanıt JSON nesnesi

- **detail** (*string*) – verbose description of the result (for HTTP status codes other than 200 OK)
- **count** (*int*) – total item count for object lists
- **next** (*string*) – next page URL for object lists
- **previous** (*string*) – previous page URL for object lists
- **results** (*array*) – results for object lists
- **url** (*string*) – URL to access this resource using API
- **web_url** (*string*) – URL to access this resource using web browser

Durum kodları

- 200 OK – when request was correctly handled
- 201 Created – when a new object was created successfully
- 204 No Content – when an object was deleted successfully
- 400 Bad Request – when form parameters are missing
- 403 Forbidden – when access is denied
- 429 Too Many Requests – when throttling is in place

Kimlik doğrulama kodları

4.10 sürümünde değişti: Project scoped tokens were introduced in the 4.10 release.

Each user has his personal access token which can be obtained in the user profile. Newly generated user tokens have the `wlu_` prefix.

It is possible to create project scoped tokens for API access to given project only. These tokens can be identified by the `wlp_` prefix.

Kimlik doğrulama örnekleri

Örnek istek:

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

Örnek yanıt:

```
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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
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 örneği:

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

Parametre aktarma örnekleri

For the **POST** method the parameters can be specified either as form submission (*application/x-www-form-urlencoded*) or as JSON (*application/json*).

Form isteği örneği:

```
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 isteği örneği:

```
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 örneği:

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

CURL JSON örneği:

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

API hız sınırlaması

The API requests are rate limited; the default configuration limits it to 100 requests per day for anonymous users and 5000 requests per hour for authenticated users.

Rate limiting can be adjusted in the `settings.py`; see [Throttling in Django REST framework documentation](#) for more details how to configure it.

In the Docker container this can be configured using `WEBLATE_API_RATELIMIT_ANON` and `WEBLATE_API_RATELIMIT_USER`.

The status of rate limiting is reported in following headers:

X-RateLimit-Limit	Yapılacak isteklerin hız sınırlama sınırı
X-RateLimit-Remaining	Kalan istek sınırı
X-RateLimit-Reset	Saniye olarak hız sınırlama penceresinin sıfırlanma süresi

4.1 sürümünde değişti: Eklenen hız sınırlaması durumu üst bilgileri.

Ayrıca bakınız:

Hız sınırlama, *Hız sınırlama*, `WEBLATE_API_RATELIMIT_ANON`, `WEBLATE_API_RATELIMIT_USER`

1.12.2 API giriş noktası

GET /api/

API kök giriş noktası.

Örnek istek:

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

Örnek yanıt:

```
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 Kullanıcılar

4.0 sürümünde geldi.

GET /api/users/

Kullanıcıları yönetme izniniz varsa kullanıcı listesini görüntüler. Yoksa, yalnızca kendi bilgilerinizi görüntüler.

Ayrıca bakınız:

Kullanıcı nesnesi öznitelikleri [GET /api/users/\(str:username\)/](#) bölümünde bulunabilir.

POST /api/users/

Yeni bir kullanıcı oluşturur.

Parametreler

- **username** (*string*) – Kullanıcı adı
- **full_name** (*string*) – Kullanıcının tam adı
- **email** (*string*) – Kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – Kullanıcı süper kullanıcı mı? (isteğe bağlı)
- **is_active** (*boolean*) – Kullanıcı etkin mi? (isteğe bağlı)
- **is_bot** (*boolean*) – Kullanıcı bot mu? (isteğe bağlı) (proje kapsamındaki belirteçler için kullanılır)

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

Kullanıcılar hakkındaki bilgileri görüntüler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **username** (*string*) – bir kullanıcının kullanıcı adı
- **full_name** (*string*) – bir kullanıcının tam adı
- **email** (*string*) – bir kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – kullanıcının süper kullanıcı olup olmadığı
- **is_active** (*boolean*) – kullanıcının etkin olup olmadığı
- **is_bot** (*boolean*) – kullanıcının bot olup olmadığı (proje kapsamındaki belirteçler için kullanılır)
- **date_joined** (*string*) – kullanıcının oluşturulduğu tarih
- **groups** (*array*) – ilişkili grup bağlantıları. Bilgi almak için: [GET /api/groups/\(int:id\)/](#)

Örnek JSON verileri:

```
{
  "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",
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

"url": "http://example.com/api/users/exampleusername/",
"statistics_url": "http://example.com/api/users/exampleusername/statistics/"
}

```

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

Kullanıcı parametrelerini değiştirir.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **username** (*string*) – bir kullanıcının kullanıcı adı
- **full_name** (*string*) – bir kullanıcının tam adı
- **email** (*string*) – bir kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – kullanıcının süper kullanıcı olup olmadığı
- **is_active** (*boolean*) – kullanıcının etkin olup olmadığı
- **is_bot** (*boolean*) – kullanıcının bot olup olmadığı (proje kapsamındaki belirteçler için kullanılır)
- **date_joined** (*string*) – kullanıcının oluşturulduğu tarih

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

Kullanıcı parametrelerini değiştirir.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **username** (*string*) – bir kullanıcının kullanıcı adı
- **full_name** (*string*) – bir kullanıcının tam adı
- **email** (*string*) – bir kullanıcının e-posta adresi
- **is_superuser** (*boolean*) – kullanıcının süper kullanıcı olup olmadığı
- **is_active** (*boolean*) – kullanıcının etkin olup olmadığı
- **is_bot** (*boolean*) – kullanıcının bot olup olmadığı (proje kapsamındaki belirteçler için kullanılır)
- **date_joined** (*string*) – kullanıcının oluşturulduğu tarih

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

Tüm kullanıcı bilgilerini siler ve kullanıcıyı etkin değil olarak işaretler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

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

Grupları bir kullanıcı ile ilişkilendirir.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı

Form parametreleri

- **string group_id** – Benzersiz grup kimliği

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

4.13.1 sürümünde geldi.

Kullanıcıyı bir gruptan kaldır.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

Form parametreleri

- **string group_id** – Benzersiz grup kimliği

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

Bir kullanıcının istatistiklerini görüntüle.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

Yanıt JSON nesnesi

- **translated** (int) – Kullanıcının yaptığı çeviri sayısı
- **suggested** (int) – Kullanıcının yaptığı öneri sayısı
- **uploaded** (int) – Kullanıcının yaptığı yükleme sayısı
- **commented** (int) – Kullanıcının yaptığı yorum sayısı
- **languages** (int) – Kullanıcının çevirebileceği dil sayısı

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

Bir kullanıcının aboneliklerini listeler.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

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

Abonelikleri bir kullanıcıyla ilişkilendirir.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı

JSON nesnesi ister

- **notification** (string) – Kayıtlı bildirimin adı
- **scope** (int) – Var olan seçeneklerden bildirim kapsamı
- **frequency** (int) – Bildirimler için sıklık seçenekleri

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

Bir kullanıcıyla ilişkilendirilmiş bir aboneliği alır.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı
- **subscription_id** (int) – Kayıtlı bildirim kimliği

PUT /api/users/ (str: username) /notifications/
int: subscription_id/

Bir kullanıcıyla ilişkili aboneliği düzenler.

Parametreler

- **username** (string) – Kullanıcının kullanıcı adı
- **subscription_id** (int) – Kayıtlı bildirim kimliği

JSON nesnesi ister

- **notification** (*string*) – Kayıtlı bildirimin adı
- **scope** (*int*) – Var olan seçeneklerden bildirim kapsamı
- **frequency** (*int*) – Bildirimler için sıklık seçenekleri

PATCH /api/users/ (**str:** *username*) /notifications/
int: *subscription_id*/

Bir kullanıcıyla ilişkili aboneliği düzenler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı
- **subscription_id** (*int*) – Kayıtlı bildirim kimliği

JSON nesnesi ister

- **notification** (*string*) – Kayıtlı bildirimin adı
- **scope** (*int*) – Var olan seçeneklerden bildirim kapsamı
- **frequency** (*int*) – Bildirimler için sıklık seçenekleri

DELETE /api/users/ (**str:** *username*) /notifications/
int: *subscription_id*/

Bir kullanıcıyla ilişkili aboneliği siler.

Parametreler

- **username** (*string*) – Kullanıcının kullanıcı adı
- **subscription_id** – Kayıtlı bildirimin adı
- **subscription_id** – tamsayı

1.12.4 Gruplar

4.0 sürümünde geldi.

GET /api/groups/

Grupları yönetme izniniz varsa grupların listesini görüntüler. Yoksa, yalnızca kullanıcının parçası olduğu grupları görüntüler.

Ayrıca bakınız:

Grup nesnesi öznetelikleri [GET /api/groups/ \(int:id\) /](#) bölümünde bulunabilir.

POST /api/groups/

Yeni bir grup oluşturur.

Parametreler

- **name** (*string*) – Grup adı
- **project_selection** (*int*) – Belirtilen seçeneklerden proje seçimi grubu
- **language_selection** (*int*) – Belirtilen seçeneklerden seçilmiş dil grubu
- **defining_project** (*str*) – *Her proje için erişim denetimi yönetimi* için kullanılan tanımlayıcı proje bağlantısı. Bilgi almak için: [GET /api/projects/ \(string:project\) /](#)

GET /api/groups/ (**int:** *id*) /

Grup hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Grubun kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir grubun adı
- **project_selection** (*int*) – proje grubuna karşılık gelen tamsayı
- **language_selection** (*int*) – dil grubuna karşılık gelen tamsayı
- **roles** (*array*) – ilişkili rollerin bağlantısı. Bilgi almak için: `GET /api/roles/(int:id)/`
- **projects** (*array*) – ilişkili projelerin bağlantısı. Bilgi almak için: `GET /api/projects/(string:project)/`
- **components** (*array*) – ilişkili bileşenlerin bağlantısı. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/`
- **componentlists** (*array*) – ilişkili bileşen listelerinin bağlantısı. Bilgi almak için: `GET /api/component-lists/(str:slug)/`
- **defining_project** (*str*) – *Her proje için erişim denetimi yönetimi için* kullanılan tanımlayıcı proje bağlantısı. Bilgi almak için: `GET /api/projects/(string:project)/`

Örnek JSON verileri:

```
{
  "name": "Guests",
  "defining_project": null,
  "project_selection": 3,
  "language_selection": 1,
  "url": "http://example.com/api/groups/1/",
  "roles": [
    "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) /`

Grup parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir grubun adı
- **project_selection** (*int*) – proje grubuna karşılık gelen tamsayı
- **language_selection** (*int*) – dil grubuna karşılık gelen tamsayı

PATCH `/api/groups/(int: id)/`

Grup parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir grubun adı
- **project_selection** (*int*) – proje grubuna karşılık gelen tamsayı
- **language_selection** (*int*) – dil grubuna karşılık gelen tamsayı

DELETE `/api/groups/(int: id)/`

Grubu siler.

Parametreler

- **id** (*int*) – Grubun kimliği

POST `/api/groups/(int: id)/roles/`

Rolleri bir gruba ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string role_id** – Benzersiz rol kimliği

POST `/api/groups/(int: id)/components/`

Bileşenleri bir gruba ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string component_id** – Benzersiz bileşen kimliği

DELETE `/api/groups/(int: id)/components/`

int: *component_id*

Bileşeni bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği
- **component_id** (*int*) – Benzersiz bileşen kimliği

POST `/api/groups/(int: id)/projects/`

Projeleri bir gruba ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string project_id** – Benzersiz proje kimliği

DELETE `/api/groups/(int: id)/projects/`

int: *project_id*

Projeyi bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği

- **project_id** (*int*) – Benzersiz proje kimliği

POST /api/groups/ (*int: id*) /languages/

Dilleri bir grupta ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string language_code** – Benzersiz dil kodu

DELETE /api/groups/ (*int: id*) /languages/
string: *language_code*

Dili bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği
- **language_code** (*string*) – Benzersiz dil kodu

POST /api/groups/ (*int: id*) /componentlists/

Bileşen listelerini bir grupta ilişkilendirir.

Parametreler

- **id** (*int*) – Grubun kimliği

Form parametreleri

- **string component_list_id** – Benzersiz bileşen listesi kimliği

DELETE /api/groups/ (*int: id*) /componentlists/
int: *component_list_id*

Bileşen listesini bir gruptan siler.

Parametreler

- **id** (*int*) – Grubun kimliği
- **component_list_id** (*int*) – Benzersiz bileşen listesi kimliği

1.12.5 Roller

GET /api/roles/

Kullanıcıyla ilişkili tüm rollerin listesini görüntüler. Kullanıcı süper kullanıcıysa, var olan tüm rollerin listesini görüntüler.

Ayrıca bakınız:

Roller nesnesi öznitelikleri [GET /api/roles/ \(*int:id*\)](#) bölümünde bulunabilir.

POST /api/roles/

Yeni bir rol oluşturur.

Parametreler

- **name** (*string*) – Rol adı
- **permissions** (*array*) – İzinlerin kod adı listesi

GET /api/roles/ (*int: id*) /

Bir rol hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Rol kimliği

Yanıt JSON nesnesi

- **name** (*string*) – Rol adı
- **permissions** (*array*) – izinlerin kod adı listesi

Örnek JSON verileri:

```
{
  "name": "Access repository",
  "permissions": [
    "vcs.access",
    "vcs.view"
  ],
  "url": "http://example.com/api/roles/1/",
}
```

PUT /api/roles/(int: id) /

Rol parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Rolün kimliği

Yanıt JSON nesnesi

- **name** (*string*) – Rol adı
- **permissions** (*array*) – izinlerin kod adı listesi

PATCH /api/roles/(int: id) /

Rol parametrelerini değiştirir.

Parametreler

- **id** (*int*) – Rolün kimliği

Yanıt JSON nesnesi

- **name** (*string*) – Rol adı
- **permissions** (*array*) – izinlerin kod adı listesi

DELETE /api/roles/(int: id) /

Rolü siler.

Parametreler

- **id** (*int*) – Rolün kimliği

1.12.6 Diller

GET /api/languages/

Tüm dillerin listesini görüntüler.

Ayrıca bakınız:Dil nesnesi öznitelikleri [GET /api/languages/\(string:language\)/](#) bölümünde bulunabilir.**POST** /api/languages/

Yeni bir dil oluşturur.

Parametreler

- **code** (*string*) – Dil adı
- **name** (*string*) – Dil adı
- **direction** (*string*) – Metin yönü

- **population** (*int*) – Konuşan sayısı
- **plural** (*object*) – Dil çoğul formülü ve sayısı

GET /api/languages/ (**string**: *language*) /

Bir dil hakkındaki bilgileri görüntüler.

Parametreler

- **language** (*string*) – Dil kodu

Yanıt JSON nesnesi

- **code** (*string*) – Dil kodu
- **direction** (*string*) – Metin yönü
- **plural** (*object*) – Dil çoğul bilgisinin nesnesi
- **aliases** (*array*) – Dil için takma adlar dizisi

JSON nesnesi ister

- **population** (*int*) – Konuşan sayısı

Örnek JSON verileri:

```
{
  "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*) /

Dil parametrelerini değiştirir.

Parametreler

- **language** (*string*) – Dilin kodu

JSON nesnesi ister

- **name** (*string*) – Dil adı
- **direction** (*string*) – Metin yönü
- **population** (*int*) – Konuşan sayısı
- **plural** (*object*) – Dil çoğul ayrıntıları

PATCH /api/languages/ (string: language) /

Dil parametrelerini değiştirir.

Parametreler

- **language** (string) – Dilin kodu

JSON nesnesi ister

- **name** (string) – Dil adı
- **direction** (string) – Metin yönü
- **population** (int) – Konuşan sayısı
- **plural** (object) – Dil çoğul ayrıntıları

DELETE /api/languages/ (string: language) /

Dili siler.

Parametreler

- **language** (string) – Dilin kodu

GET /api/languages/ (string: language) /statistics/

Bir dilin istatistiklerini görüntüler.

Parametreler

- **language** (string) – Dil kodu

Yanıt JSON nesnesi

- **total** (int) – toplam dizge sayısı
- **total_words** (int) – toplam sözcük sayısı
- **last_change** (timestamp) – dildeki son değişiklikler
- **recent_changes** (int) – toplam değişiklik sayısı
- **translated** (int) – çevrilmiş dizge sayısı
- **translated_percent** (float) – çevrilmiş dizge yüzdesi
- **translated_words** (int) – çevrilmiş sözcük sayısı
- **translated_words_percent** (int) – çevrilmiş sözcük yüzdesi
- **translated_chars** (int) – çevrilmiş karakter sayısı
- **translated_chars_percent** (int) – çevrilmiş karakter yüzdesi
- **total_chars** (int) – toplam karakter sayısı
- **fuzzy** (int) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **fuzzy_percent** (int) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **failing** (int) – denetim hatası olan dizge sayısı
- **failing** – denetim hatası olan dizge yüzdesi

1.12.7 Projeler

GET /api/projects/

Tüm projelerin listesini görüntüler.

Ayrıca bakınız:

Proje nesnesi öznetelikleri *GET /api/projects/(string:project)/* bölümünde bulunabilir.

POST /api/projects/

3.9 sürümünde geldi.

Yeni bir proje oluşturur.

Parametreler

- **name** (*string*) – Proje adı
- **slug** (*string*) – Proje kısaltması
- **web** (*string*) – Proje web sitesi

GET /api/projects/(string: project) /

Bir proje hakkındaki bilgileri görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **name** (*string*) – proje adı
- **slug** (*string*) – proje kısaltması
- **web** (*string*) – proje web sitesi
- **components_list_url** (*string*) – Bileşen listesi adresi. Bilgi almak için: *GET /api/projects/(string:project)/components/*
- **repository_url** (*string*) – Depo durumu adresi. Bilgi almak için: *GET /api/projects/(string:project)/repository/*
- **changes_list_url** (*string*) – Değişiklikler listesi adresi. Bilgi almak için: *GET /api/projects/(string:project)/changes/*
- **translation_review** (*boolean*) – *Onaylama kullanılsın*
- **source_review** (*boolean*) – *Kaynak onaylama kullanılsın*
- **set_language_team** (*boolean*) – *“Language-Team” üst bilgisi ayarlansın*
- **enable_hooks** (*boolean*) – *Kancalar kullanılsın*
- **instructions** (*string*) – *Çeviri yönergeleri*
- **language_aliases** (*string*) – *Dil kod adları*

Örnek JSON verileri:

```
{
  "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) /`

4.3 sürümünde geldi.

Bir projeyi **PATCH** isteğiyle düzenler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

PUT `/api/projects/(string: project) /`

4.3 sürümünde geldi.

Bir projeyi **PUT** isteğiyle düzenler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

DELETE `/api/projects/(string: project) /`

3.9 sürümünde geldi.

Bir projeyi siler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

GET `/api/projects/(string: project) /changes/`

Returns a list of project changes. This is essentially a project scoped **GET** `/api/changes/` accepting same params.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – array of component objects; see **GET** `/api/changes/(int:id)/`

GET `/api/projects/(string: project) /repository/`

Returns information about VCS repository status. This endpoint contains only an overall summary for all repositories for the project. To get more detailed status use **GET** `/api/components/(string:project)/(string:component)/repository/`.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **needs_commit** (*boolean*) – whether there are any pending changes to commit
- **needs_merge** (*boolean*) – whether there are any upstream changes to merge
- **needs_push** (*boolean*) – whether there are any local changes to push

Örnek JSON verileri:

```
{
  "needs_commit": true,
  "needs_merge": false,
  "needs_push": true
}
```

POST /api/projects/(string: project)/repository/

Performs given operation on the VCS repository.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

JSON nesnesi ister

- **operation** (*string*) – Operation to perform: one of push, pull, commit, re-set, cleanup, file-sync

Yanıt JSON nesnesi

- **result** (*boolean*) – result of the operation

CURL örneği:

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

JSON isteği örneği:

```
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 yanıtı örneği:

```
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/

Belirtilen projedeki çeviri bileşenlerinin listesini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – array of component objects; see [GET /api/components/\(string:project\)/\(string:component\)/](#)

POST /api/projects/(string: project)/components/

3.9 sürümünde geldi.

4.3 sürümünde değişti: The zipfile and docfile parameters are now accepted for VCS-less components, see [Yerel dosyalar](#).

4.6 sürümünde değişti: The cloned repositories are now automatically shared within a project using [İç Weblate adresleri](#). Use `disable_autoshare` to turn off this.

Belirtilen projede çeviri bileşenleri oluşturur.

İpucu: Use *İç Weblate adresleri* when creating multiple components from a single VCS repository.

Not: Most of the component creation happens in the background. Check the `task_url` attribute of created component and follow the progress there.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Form parametreleri

- **file zipfile** – ZIP file to upload into Weblate for translations initialization
- **file docfile** – Çevrilecek belge
- **boolean disable_autoshare** – Disables automatic repository sharing via *İç Weblate adresleri*.

JSON nesnesi ister

- **object** – Bileşen parametreleri. Ayrıntılı bilgi almak için `GET /api/components/(string:project)/(string:component)/`

Yanıt JSON nesnesi

- **result** (*object*) – Created component object; see `GET /api/components/(string:project)/(string:component)/`

JSON can not be used when uploading the files using the `zipfile` and `docfile` parameters. The data has to be uploaded as *multipart/form-data*.

CURL form isteği örneği:

```
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 isteği örneği:

```
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/
```

Git üzerinden yeni bir bileşen oluşturan JSON isteği:

```
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"
}
```

Başka bir bileşenden yeni bir bileşen oluşturan JSON isteği:

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

{
  "file_format": "po",
  "filemask": "po/*.po",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "weblate://weblate/hello",
  "template": "",
  "new_base": "po/hello.pot",
  "vcs": "git"
}
```

JSON yanıtı örneği:

```
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": {
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "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/`

Bir projedeki tüm diller için sayılanmış istatistikleri görüntüler.

3.8 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – çeviri istatistiği nesneleri dizisi
- **language** (*string*) – dil adı
- **code** (*string*) – dil kodu
- **total** (*int*) – toplam dizge sayısı
- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **total_words** (*int*) – toplam sözcük sayısı
- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **words_percent** (*float*) – çevrilmiş sözcük yüzdesi

GET `/api/projects/(string: project)/statistics/`

Bir projenin istatistiklerini görüntüler.

3.8 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması

Yanıt JSON nesnesi

- **total** (*int*) – toplam dizge sayısı
- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **total_words** (*int*) – toplam sözcük sayısı

- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **words_percent** (*float*) – çevrilmiş sözcük yüzdesi

1.12.8 Bileşenler

İpucu: Yeni bileşenler oluşturmak için `POST /api/projects/(string:project)/components/` kullanın.

GET /api/components/

Çeviri bileşenlerinin listesini görüntüler.

Ayrıca bakınız:

Bileşen nesnesi özenitelikleri `GET /api/components/(string:project)/(string:component)/` bölümünde bulunabilir.

GET /api/components/(string: project) / string: component/

Çeviri bileşeni hakkındaki bilgileri görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **project** (*object*) – çeviri projesi. Bilgi almak için `GET /api/projects/(string:project)/`
- **name** (*string*) – *Bileşen adı*
- **slug** (*string*) – *Bileşen kısaltması*
- **vcs** (*string*) – *Sürüm denetimi sistemi*
- **repo** (*string*) – *Kaynak kod deposu*
- **git_export** (*string*) – *Dışa aktarılmış depo adresi*
- **branch** (*string*) – *Depo dalı*
- **push_branch** (*string*) – *İtme işleminin yapılacağı dal*
- **filemask** (*string*) – *Dosya maskesi*
- **template** (*string*) – *Tek dilli temel dil dosyası*
- **edit_template** (*string*) – *Temel dosyayı düzenle*
- **intermediate** (*string*) – *Ara dil dosyası*
- **new_base** (*string*) – *Yeni çeviriler kalıbı*
- **file_format** (*string*) – *Dosya biçimi*
- **license** (*string*) – *Çeviri lisansı*
- **agreement** (*string*) – *Katılımcı anlaşması*
- **new_lang** (*string*) – *Yeni çeviri ekleme*
- **language_code_style** (*string*) – *Dil kodu biçimi*
- **source_language** (*object*) – kaynak dil nesnesi. Ayrıntılı bilgi alın `GET /api/languages/(string:language)/`

- **push** (*string*) – *Depo itme adresi*
- **check_flags** (*string*) – *Çeviri işaretleri*
- **priority** (*string*) – *Öncelik*
- **enforced_checks** (*string*) – *Zorunlu denetimler*
- **restricted** (*string*) – *Kısıtlı erişim*
- **repoweb** (*string*) – *Depo tarayıcısı*
- **report_source_bugs** (*string*) – *Kaynak dizge hata bildirme adresi*
- **merge_style** (*string*) – *Birleştirme biçemi*
- **commit_message** (*string*) – *İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri*
- **add_message** (*string*) – *İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri*
- **delete_message** (*string*) – *İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri*
- **merge_message** (*string*) – *İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri*
- **addon_message** (*string*) – *İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri*
- **pull_message** (*string*) – *İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri*
- **allow_translation_propagation** (*string*) – *Çevirilerin yayılmasını sağlamak*
- **enable_suggestions** (*string*) – *Öneriler kullanılsın*
- **suggestion_voting** (*string*) – *Öneri oylaması*
- **suggestion_autoaccept** (*string*) – *Öneriler kendiliğinden kabul edilsin*
- **push_on_commit** (*string*) – *İşleme ile itme*
- **commit_pending_age** (*string*) – *İşlenecek değişikliklerin yaşı*
- **auto_lock_error** (*string*) – *Hata durumunda kilitleme*
- **language_regex** (*string*) – *Dil süzgeci*
- **variant_regex** (*string*) – *Çeşitler kurallı ifadesi*
- **repository_url** (*string*) – Depo durumu adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – Çeviri listesi adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – Kilit durumu adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – Değişiklik listesi adresi. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – Arka plan görevi (varsa) adresi. Bilgi almak için: `GET /api/tasks/(str:uuid)/`

Örnek JSON verileri:

```
{
  "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",
    "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/

Bir bileşeni **PATCH** isteği ile düzenleme.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması
- **source_language** (string) – Proje kaynak dili kodu (isteğe bağlı)

JSON nesnesi ister

- **name** (string) – bileşen adı
- **slug** (string) – bileşen kısaltması
- **repo** (string) – Sürüm denetimi sistemi deposunun adresi

CURL örneği:

```
curl \
  --data-binary '{"name": "new name"}' \
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
-H "Content-Type: application/json" \
-H "Authorization: Token TOKEN" \
PATCH http://example.com/api/projects/hello/components/
```

JSON isteği örneği:

```
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 yanıtı örneği:

```
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": "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/
```

Bir bileşeni [PUT](#) isteği ile düzenleme.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **branch** (*string*) – Sürüm denetimi sistemi deposunun dalı
- **file_format** (*string*) – çevirilerin dosya biçimi
- **filemask** (*string*) – depodaki çeviri dosyalarının modeli
- **name** (*string*) – bileşen adı
- **slug** (*string*) – bileşen kısaltması
- **repo** (*string*) – Sürüm denetimi sistemi deposunun adresi
- **template** (*string*) – tek dilli çeviriler için temel dosya
- **new_base** (*string*) – yeni çeviriler eklemek için temel dosya
- **vcs** (*string*) – sürüm denetimi sistemi

DELETE /api/components/ (**string:** *project*) /
string: *component* /

3.9 sürümünde geldi.

Bir bileşeni siler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

GET /api/components/ (**string:** *project*) /
string: *component/changes/*

Bileşen değişikliklerinin listesini görüntüler. Temelde [GET /api/changes/](#) kapsamında bir bileşendir ve aynı parametreleri kabul eder.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – array of component objects; see [GET /api/changes/ \(int:id\) /](#)

GET /api/components/ (**string:** *project*) /
string: *component/file/*

4.9 sürümünde geldi.

Bileşenle ilişkili tüm kullanılabilir çevirileri istenilen biçimdeki arşiv dosyası olarak indirir.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Sorgu parametreleri

- **format** (*string*) – Kullanılacak arşiv biçimi belirtilmemişse varsayılan olarak zip biçimi kullanılır. Desteklenen biçimler: zip

GET `/api/components/(string: project) /string: component/screenshots/`

Bileşen ekran görüntülerinin listesini görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – bileşen ekran görüntüleri dizisi. Bilgi almak için: `GET /api/screenshots/(int:id)/`

GET `/api/components/(string: project) /string: component/lock/`

Bileşenin kilit durumunu görüntüler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **locked** (*boolean*) – bileşenin güncellemeler için kilitlenip kilitlenmediği

Örnek JSON verileri:

```
{
  "locked": false
}
```

POST `/api/components/(string: project) /string: component/lock/`

Bileşenin kilit durumunu ayarlar.

Yanıt `GET /api/components/(string:project)/(string:component)/lock/` ile aynıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **lock** – Kilitli olduğunu ya da olmadığını belirten ikili değer.

CURL örneği:

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

JSON isteği örneği:

```
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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
{"lock": true}
```

JSON yanıtı örneği:

```
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/`

Sürüm denetimi sistemi deposunun durumu hakkındaki bilgileri görüntüler.

Yanıt `GET /api/projects/(string:project)/repository/` ile aynıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **needs_commit** (*boolean*) – whether there are any pending changes to commit
- **needs_merge** (*boolean*) – whether there are any upstream changes to merge
- **needs_push** (*boolean*) – whether there are any local changes to push
- **remote_commit** (*string*) – Uzaktan işleme bilgileri
- **status** (*string*) – VCS repository status as reported by VCS
- **merge_failure** – Text describing merge failure or null if there is none

POST `/api/components/(string: project) /`
string: `component/repository/`

Performs the given operation on a VCS repository.

See `POST /api/projects/(string:project)/repository/` for documentation.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **operation** (*string*) – Operation to perform: one of push, pull, commit, re-set, cleanup

Yanıt JSON nesnesi

- **result** (*boolean*) – result of the operation

CURL örneği:

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

JSON isteği örneği:

```
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 yanıtı örneği:

```
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/
Tek dilli çeviriler için temel dosyayı indirir.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması

GET /api/components/(string: project) /
string: component/new_template/
Yeni çeviriler için kalıp dosyasını indirir.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması

GET /api/components/(string: project) /
string: component/translations/
Belirtilen bileşendeki çeviri nesnelerinin listesini görüntüler.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (array) – array of translation objects; see `GET /api/translations/(string:project)/(string:component)/(string:language)/`

POST /api/components/(string: project) /
string: component/translations/

Belirtilen bileşende yeni çeviri oluşturur.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

JSON nesnesi ister

- **language_code** (*string*) – translation language code; see `GET /api/languages/(string:language)/`

Yanıt JSON nesnesi

- **result** (*object*) – new translation object created

CURL örneği:

```
curl \
  -d language_code=cs \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

JSON isteği örneği:

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

{"language_code": "cs"}
```

JSON yanıtı örneği:

```
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/",
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "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/`

Returns paginated statistics for all translations within component.

2.7 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **results** (*array*) – array of translation statistics objects; see `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`

GET `/api/components/(string: project) /`
string: `component/links/`

Bir bileşenle ilişkili projeleri görüntüler.

4.5 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Yanıt JSON nesnesi

- **projects** (*array*) – ilişkili projeler. Bilgi almak için: `GET /api/projects/(string:project)/`

POST `/api/components/(string: project) /`
string: `component/links/`

Projeyi bir bileşenle ilişkilendirir.

4.5 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması

Form parametreleri

- **string project_slug** – Proje kısaltması

DELETE /api/components/ (string: project) /
string: component/links/string: project_slug/

Bir projenin bir bileşenle ilişkisini kaldırır.

4.5 sürümünde geldi.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması
- **project_slug** (string) – Kaldırılacak projenin kısaltması

1.12.9 Çeviriler

GET /api/translations/

Çevirilerin listesini görüntüler.

Ayrıca bakınız:

Çeviri nesnesi öznitelikleri **GET** /api/translations/(string:project)/
(string:component)/(string:language)/ bölümünde bulunabilir.

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

Bir çeviriyle ilgili bilgileri görüntüler.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması
- **language** (string) – Çeviri dili kodu

Yanıt JSON nesnesi

- **component** (object) – bileşen nesnesi. Bilgi almak için: **GET** /api/components/(string:project)/(string:component)/
- **failing_checks** (int) – denetim hatası olan dizge sayısı
- **failing_checks_percent** (float) – denetim hatası olan dizge yüzdesi
- **failing_checks_words** (int) – denetim hatası olan sözcük sayısı
- **filename** (string) – çeviri dosyası adı
- **fuzzy** (int) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **fuzzy_percent** (float) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **fuzzy_words** (int) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **have_comment** (int) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **have_suggestion** (int) – önerileri olan dizge sayısı
- **is_template** (boolean) – çeviri tabanının tek dilli olup olmadığı
- **language** (object) – kaynak dil nesnesi. Ayrıntılı bilgi alın **GET** /api/languages/(string:language)/
- **language_code** (string) – depoda kullanılan dil kodu. Dil nesnesindeki dil kodundan farklı olabilir
- **last_author** (string) – son yazarın adı
- **last_change** (timestamp) – son değişiklik zaman damgası

- **revision** (*string*) – dosyanın sürüm karması
- **share_url** (*string*) – Etkileşim sayfasına yönlendiren paylaşım adresi
- **total** (*int*) – toplam dizge sayısı
- **total_words** (*int*) – toplam sözcük sayısı
- **translate_url** (*string*) – Çeviri adresi
- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **repository_url** (*string*) – Depo durumu adresi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
- **file_url** (*string*) – Dosya nesnesi adresi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/file/`
- **changes_list_url** (*string*) – Değişiklik listesi adresi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/changes/`
- **units_list_url** (*string*) – Dizge listesi adresi. Bilgi almak için: `GET /api/translations/(string:project)/(string:component)/(string:language)/units/`

Örnek JSON verileri:

```
{
  "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/"
  },
}
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

"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,
"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/`

3.9 sürümünde geldi.

Bir çeviriyi siler.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

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

Çeviri değişikliklerinin listesini görüntüler. Temelde aynı parametreleri kabul eden bir `GET /api/changes/` çeviri kapsamıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

Yanıt JSON nesnesi

- **results** (*array*) – array of component objects; see `GET /api/changes/(int:id) /`

GET /api/translations/ (string: project) /
string: component/string: language/units/

Çeviri birimlerinin listesini görüntüler.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması
- **language** (string) – Çeviri dili kodu
- **q** (string) – Arama sorgusu dizgesi *Arama* (isteğe bağlı)

Yanıt JSON nesnesi

- **results** (array) – bileşen nesneleri dizisi. Bilgi almak için: `GET /api/units/(int:id)/`

POST /api/translations/ (string: project) /
string: component/string: language/units/

Yeni birim ekle.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması
- **language** (string) – Çeviri dili kodu

JSON nesnesi ister

- **key** (string) – Çeviri biriminin adı (anahtar veya bağlam olarak kullanılır)
- **value** (array) – Kaynak dizgeler (çoğul oluşturmuyorsanız tekil dizge kullanın)
- **state** (int) – Dizge durumu. Bilgi almak için: `GET /api/units/(int:id)/`

Yanıt JSON nesnesi

- **unit** (object) – yeni oluşturulmuş birim. Bilgi almak için: `GET /api/units/(int:id)/`

Ayrıca bakınız:

Dizge yönetimi, adding-new-strings

POST /api/translations/ (string: project) /
string: component/string: language/autotranslate/

Kendiliğinden çeviriyi tetikler.

Parametreler

- **project** (string) – Proje adresi kısaltması
- **component** (string) – Bileşen adresinin kısaltması
- **language** (string) – Çeviri dili kodu

JSON nesnesi ister

- **mode** (string) – Kendiliğinden çeviri kipi
- **filter_type** (string) – Kendiliğinden çeviri süzgeci türü
- **auto_source** (string) – Kendiliğinden çeviri kaynağı - mt ya da others
- **component** (string) – Ek bileşenlere erişebilmesi için projenin paylaşılan çeviri bel-
leğine katkıda bulunmayı etkinleştirin.
- **engines** (array) – Makine çevirisi hizmetleri

- **threshold** (*string*) – Skor eşiği

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

Download current translation file as it is stored in the VCS (without the `format` parameter) or converted to another format (see [Downloading translations](#)).

Not: This API endpoint uses different logic for output than rest of API as it operates on whole file rather than on data. Set of accepted `format` parameter differs and without such parameter you get translation file as stored in VCS.

Sorgu parametreleri

- **format** – File format to use; if not specified no format conversion happens; supported file formats: `po`, `mo`, `xliff`, `xliff11`, `tbx`, `tmx`, `csv`, `xlsx`, `json`, `aresource`, `strings`
- **q** (*string*) – Filter downloaded strings, see `search`, only applicable when conversion is in place (`format` is specified).

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

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

Upload new file with translations.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

Form parametreleri

- **string conflicts** – How to deal with conflicts (`ignore`, `replace-translated` or `replace-approved`)
- **file file** – Yüklenen dosya
- **string email** – İlgilinin e-posta adresi
- **string author** – Katkıda bulunanın adı
- **string method** – Yükleme yöntemi (`translate`, `approve`, `suggest`, `fuzzy`, `replace`, `source`, `add`). Bilgi almak için: [Import methods](#)
- **string fuzzy** – Belirsiz (düzenlenecek olarak imlenmiş) dizgeleri işleme (`empty`, `process`, `approve`)

CURL örneği:

```
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**/

Sürüm denetimi sistemi deposunun durumu hakkındaki bilgileri görüntüler.

Yanıt `GET /api/components/(string:project)/(string:component)/repository/` ile aynıdır.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

**POST /api/translations/(string: project) /
string: component/string: language/repository/**

Performs given operation on the VCS repository.

See `POST /api/projects/(string:project)/repository/` for documentation.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

JSON nesnesi ister

- **operation** (*string*) – Operation to perform: one of push, pull, commit, reset, cleanup

Yanıt JSON nesnesi

- **result** (*boolean*) – result of the operation

**GET /api/translations/(string: project) /
string: component/string: language/statistics/**

Ayrıntılı çeviri istatistiklerini görüntüler.

2.7 sürümünde geldi.

Parametreler

- **project** (*string*) – Proje adresi kısaltması
- **component** (*string*) – Bileşen adresinin kısaltması
- **language** (*string*) – Çeviri dili kodu

Yanıt JSON nesnesi

- **code** (*string*) – dil kodu
- **failing** (*int*) – denetim hataları sayısı
- **failing_percent** (*float*) – denetim hataları yüzdesi
- **fuzzy** (*int*) – belirsiz (düzenlenecek olarak imlenmiş) dizge sayısı
- **fuzzy_percent** (*float*) – belirsiz (düzenlenecek olarak imlenmiş) dizge yüzdesi
- **total_words** (*int*) – toplam sözcük sayısı
- **translated_words** (*int*) – çevrilmiş sözcük sayısı
- **last_author** (*string*) – son yazarın adı
- **last_change** (*timestamp*) – son değişiklik tarihi
- **name** (*string*) – dil adı
- **total** (*int*) – toplam dizge sayısı

- **translated** (*int*) – çevrilmiş dizge sayısı
- **translated_percent** (*float*) – çevrilmiş dizge yüzdesi
- **url** (*string*) – Çeviriye erişme adresi (etkileşim adresi)
- **url_translate** (*string*) – Çeviriye erişme adresi (gerçek çeviri adresi)

1.12.10 Bellek

4.14 sürümünde geldi.

GET /api/memory/

Bileşen listelerinin listesini görüntüler.

DELETE /api/memory/ (*int: memory_object_id*) /

Bir çeviri belleği nesnesini siler

Parametreler

- **memory_object_id** – Çeviri belleği nesnesi kimliği

1.12.11 Birimler

A *unit* is a single piece of a translation which pairs a source string with a corresponding translated string and also contains some related metadata. The term is derived from the [Translate Toolkit](#) and XLIFF.

2.10 sürümünde geldi.

GET /api/units/

Çeviri birimlerinin listesini görüntüler.

Parametreler

- **q** (*string*) – Arama sorgusu dizgesi [Arama](#) (isteğe bağlı)

Ayrıca bakınız:

Birim nesnesi öznitelikleri [GET /api/units/\(int:id\)/](#) bölümünde bulunabilir.

GET /api/units/ (*int: id*) /

4.3 sürümünde değişti: Hedef ve kaynak çoğul dizgeleri doğru işlemek için diziye dönüştürüldü.

Çeviri birimi hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Birim kimliği

Yanıt JSON nesnesi

- **translation** (*string*) – İlişkili çeviri nesnesinin adresi
- **source** (*array*) – kaynak dizge
- **previous_source** (*string*) – belirsiz eşleştirmesi için kullanılan önceki kaynak dizge
- **target** (*array*) – hedef dizge
- **id_hash** (*string*) – birimin benzersiz kimliği
- **content_hash** (*string*) – kaynak dizgenin benzersiz kimliği
- **location** (*string*) – birimin kaynak kodundaki konumu
- **context** (*string*) – çeviri birimi bağlamı
- **note** (*string*) – çeviri birimi notu

- **flags** (*string*) – çeviri birimi imleri
- **labels** (*array*) – çeviri birimi etiketleri, kaynak birimlerde kullanılabilir
- **state** (*int*) – birim durumu, 0 - Çevrilmemiş, 10 - Düzenlenmesi gerekli, 20 - Çevrilmiş, 30 - Onaylanmış, 100 - Salt okunur
- **fuzzy** (*boolean*) – birimin belirsiz ya da onaylanmayı bekliyor olarak imlenmiş olup olmadığı
- **translated** (*boolean*) – birimin çevrilmiş olup olmadığı
- **approved** (*boolean*) – çevirinin onaylanmış olup olmadığı
- **position** (*int*) – birimin çeviri dosyasındaki konumu
- **has_suggestion** (*boolean*) – birimin önerileri olup olmadığı
- **has_comment** (*boolean*) – birimin yorumları olup olmadığı
- **has_failing_check** (*boolean*) – birimde denetim hatası olup olmadığı
- **num_words** (*int*) – kaynak sözcük sayısı
- **priority** (*int*) – çeviri önceliği. Varsayılan değer: 100
- **id** (*int*) – birim kimliği
- **explanation** (*string*) – Dizge açıklaması. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Additional info on source strings](#)
- **extra_flags** (*string*) – Ek dizge imleri. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)
- **web_url** (*string*) – Birimin düzenlenebileceği adres
- **source_unit** (*string*) – Kaynak birim bağlantısı. Bilgi almak için: `GET /api/units/(int:id)/`
- **pending** (*boolean*) – birimin belirsiz ya da onaylanmış olarak imlenmiş olup olmadığı
- **timestamp** (*timestamp*) – dizge yaşı

PATCH /api/units/(int: id) /

4.3 sürümünde geldi.

Çeviri birimi üzerinde bazı güncellemeler yapar.

Parametreler

- **id** (*int*) – Birim kimliği

JSON nesnesi ister

- **state** (*int*) – birim durumu, 0 - Çevrilmemiş, 10 - Düzenlenmesi gerekli, 20 - Çevrilmiş, 30 - Onaylanmış (gözden geçirme iş akışı etkinleştirilmelidir. Bilgi almak için: `:ref: reviews`)
- **target** (*array*) – hedef dizge
- **explanation** (*string*) – Dizge açıklaması. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Additional info on source strings](#)
- **extra_flags** (*string*) – Ek dizge imleri. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)

Yanıt JSON nesnesi

- **labels** (*array*) – etiketler, kaynak birimlerde kullanılabilir

PUT `/api/units/(int: id) /`

4.3 sürümünde geldi.

Çeviri birimi üzerinde tüm güncellemeleri yapar.

Parametreler

- **id** (*int*) – Birim kimliği

JSON nesnesi ister

- **state** (*int*) – birim durumu, 0 - Çevrilmemiş, 10 - Düzenlenmesi gerekli, 20 - Çevrilmiş, 30 - Onaylanmış (gözden geçirme iş akışı etkinleştirilmelidir. Bilgi almak için: [:ref: reviews](#))
- **target** (*array*) – hedef dizge
- **explanation** (*string*) – Dizge açıklaması. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Additional info on source strings](#)
- **extra_flags** (*string*) – Ek dizge imleri. Kaynak birimlerde kullanılabilir. Bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)

Yanıt JSON nesnesi

- **labels** (*array*) – etiketler, kaynak birimlerde kullanılabilir

DELETE `/api/units/(int: id) /`

4.3 sürümünde geldi.

Bir çeviri birimini siler.

Parametreler

- **id** (*int*) – Birim kimliği

1.12.12 Değişiklikler

2.10 sürümünde geldi.

GET `/api/changes/`

4.1 sürümünde değişti: Değişiklikleri süzme özelliği 4.1 sürümünde sunuldu.

Çeviri değişikliklerinin listesini görüntüler.

Ayrıca bakınız:

Değişiklik nesnesi öznitelikleri [GET /api/changes/\(int:id\) /](#) bölümünde bulunabilir.

Sorgu parametreleri

- **user** (*string*) – Süzülecek kullanıcının kullanıcı adı
- **action** (*int*) – Süzülecek işlem, birkaç kez kullanılabilir
- **timestamp_after** (*timestamp*) – Sonraki değişiklikleri listelemek için ISO 8601 biçiminde zaman damgası
- **timestamp_before** (*timestamp*) – Önceki değişiklikleri listelemek için ISO 8601 biçiminde zaman damgası

GET `/api/changes/(int: id) /`

Çeviri değişikliği hakkındaki bilgileri görüntüler.

Parametreler

- **id** (*int*) – Değişiklik kimliği

Yanıt JSON nesnesi

- **unit** (*string*) – İlişkili birim nesnesinin adresi
- **translation** (*string*) – İlişkili çeviri nesnesinin adresi
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **user** (*string*) – İlişkili kullanıcı nesnesinin adresi
- **author** (*string*) – İlişkili yazar nesnesinin adresi
- **timestamp** (*timestamp*) – etkinlik zaman damgası
- **action** (*int*) – işlemin sayısal kimliği
- **action_name** (*string*) – işlemin metin açıklaması
- **target** (*string*) – metin ya da ayarın değışiklięi etkinlięi
- **id** (*int*) – değışiklik kimlięi

1.12.13 Ekran görüntüleri

2.14 sürümünde geldi.

GET /api/screenshots/

Ekran görüntüsü dizgesi bilgilerinin listesini görüntüler.

Ayrıca bakınız:

Ekran görüntüsü nesnesi öznitelikleri [GET /api/screenshots/\(int:id\)/](#) bölümünde bulunabilir.

GET /api/screenshots/(int: id) /

Ekran görüntüsü bilgileriyle ilgili bilgileri görüntüler.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimlięi

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: [GET /api/screenshots/\(int:id\)/file/](#)
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: [GET /api/units/\(int:id\)/](#)

GET /api/screenshots/(int: id) /file/

Ekran görüntüsünü indir.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimlięi

POST /api/screenshots/(int: id) /file/

Ekran görüntüsünü değıştir.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimlięi

Form parametreleri

- **file image** – Yüklenen dosya

CURL örneęi:

```
curl -X POST \
  -F image=@image.png \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/screenshots/1/file/
```

POST /api/screenshots/(int: id)/units/

Kaynak dizgeyi ekran görüntüsüyle ilişkilendirir.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Form parametreleri

- **string unit_id** – Birim kimliği

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **translation** (*string*) – İlişkili çeviri nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: `GET /api/units/(int:id)/`

DELETE /api/screenshots/(int: id)/units/ int: unit_id

Kaynak dizge ile ekran görüntüsünün ilişkilendirmesini kaldırır.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği
- **unit_id** – Kaynak dizge birimi kimliği

POST /api/screenshots/

Yeni bir ekran görüntüsü oluşturur.

Form parametreleri

- **file image** – Yüklenen dosya
- **string name** – Ekran görüntüsü adı
- **string project_slug** – Proje kısaltması
- **string component_slug** – Bileşen kısaltması
- **string language_code** – Dil kodu

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: `GET /api/units/(int:id)/`

PATCH /api/screenshots/(int: id)/

Ekran görüntüsü ile ilgili bazı bilgileri düzenle.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: `GET /api/units/(int:id)/`

PUT /api/screenshots/(int: id) /

Ekran görüntüsü ile ilgili tüm bilgileri düzenle.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

Yanıt JSON nesnesi

- **name** (*string*) – ekran görüntüsünün adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **file_url** (*string*) – Bir dosyayı indirme bağlantısı. Bilgi almak için: `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – ilişkili kaynak dizge bilgileri bağlantısı. Bilgi almak için: `GET /api/units/(int:id)/`

DELETE /api/screenshots/(int: id) /

Ekran görüntüsünü sil.

Parametreler

- **id** (*int*) – Ekran görüntüsü kimliği

1.12.14 Eklentiler

4.4.1 sürümünde geldi.

GET /api/addons/

Eklentilerin listesini görüntüler.

Ayrıca bakınız:

Eklenti nesnesi öznelikleri `GET /api/addons/(int:id)/` bölümünde bulunabilir.

GET /api/addons/(int: id) /

Eklentiyle ilgili bilgileri görüntüler.

Parametreler

- **id** (*int*) – Eklenti kimliği

Yanıt JSON nesnesi

- **name** (*string*) – bir eklentinin adı
- **component** (*string*) – İlişkili bileşen nesnesinin adresi
- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

Ayrıca bakınız:

Eklentiler

POST /api/components/(string: project) /
string: component/addons/

Yeni bir eklenti oluşturur.

Parametreler

- **project_slug** (*string*) – Proje kısaltması
- **component_slug** (*string*) – Bileşen kısaltması

JSON nesnesi ister

- **name** (*string*) – bir eklentinin adı
- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

PATCH /api/addons/ (int: id) /

Eklenti ile ilgili bazı bilgileri düzenle.

Parametreler

- **id** (*int*) – Eklenti kimliği

Yanıt JSON nesnesi

- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

PUT /api/addons/ (int: id) /

Eklenti ile ilgili tüm bilgileri düzenle.

Parametreler

- **id** (*int*) – Eklenti kimliği

Yanıt JSON nesnesi

- **configuration** (*object*) – İsteğe bağlı eklenti yapılandırması

DELETE /api/addons/ (int: id) /

Eklentiği sil.

Parametreler

- **id** (*int*) – Eklenti kimliği

1.12.15 Bileşen listeleri

4.0 sürümünde geldi.

GET /api/component-lists/

Bileşen listelerinin listesini görüntüler.

Ayrıca bakınız:

Bileşen listesi nesnesi öznitelikleri [GET /api/component-lists/\(str:slug\)/](#) bölümünde bulunabilir.

GET /api/component-lists/ (str: slug) /

Bileşen listesi hakkındaki bilgileri görüntüler.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

Yanıt JSON nesnesi

- **name** (*string*) – bir bileşen listesinin adı
- **slug** (*string*) – bir bileşen listesinin kısaltması
- **show_dashboard** (*boolean*) – bir panoda görüntülenip görüntülenmeyeceği

- **components** (*array*) – ilişkili bileşenlerin bağlantısı. Bilgi almak için: `GET /api/components/(string:project)/(string:component)/`
- **auto_assign** (*array*) – kendiliğinden atama kuralları

PUT /api/component-lists/ (str: slug) /

Bileşen listesi parametrelerini değiştirir.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

JSON nesnesi ister

- **name** (*string*) – bir bileşen listesinin adı
- **slug** (*string*) – bir bileşen listesinin kısaltması
- **show_dashboard** (*boolean*) – bir panoda görüntülenip görüntülenmeyeceği

PATCH /api/component-lists/ (str: slug) /

Bileşen listesi parametrelerini değiştirir.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

JSON nesnesi ister

- **name** (*string*) – bir bileşen listesinin adı
- **slug** (*string*) – bir bileşen listesinin kısaltması
- **show_dashboard** (*boolean*) – bir panoda görüntülenip görüntülenmeyeceği

DELETE /api/component-lists/ (str: slug) /

Bileşen listesini siler.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

POST /api/component-lists/ (str: slug) /components/

Bileşeni bir bileşen listesiyle ilişkilendirir.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması

Form parametreleri

- **string component_id** – Bileşen kimliği

DELETE /api/component-lists/ (str: slug) /components/

str: component_slug

Bileşen listesinden bir bileşenin ilişkisini kaldırır.

Parametreler

- **slug** (*string*) – Bileşen listesi kısaltması
- **component_slug** (*string*) – Bileşen kısaltması

1.12.16 Sözlük

4.5 sürümünde değişti: Sözlükler artık normal bileşenler, çeviriler ve dizgeler olarak kaydediliyor. Lütfen bunun yerine ilgili API uygulamasını kullanın.

1.12.17 Görevler

4.4 sürümünde geldi.

GET /api/tasks/

Görevlerin listesi şu anda alınamıyor.

GET /api/tasks/(str: uuid) /

Bir görev hakkındaki bilgileri görüntüler

Parametreler

- **uuid** (*string*) – Görev UUID

Yanıt JSON nesnesi

- **completed** (*boolean*) – Görevin tamamlanıp tamamlanmadığı
- **progress** (*int*) – Yüzde olarak görev ilerlemesi
- **result** (*object*) – Görev sonucu veya ilerleme ayrıntıları
- **log** (*string*) – Görev günlüğü

1.12.18 Ölçümler

GET /api/metrics/

Sunucu ölçümlerini görüntüler.

Yanıt JSON nesnesi

- **units** (*int*) – Birim sayısı
- **units_translated** (*int*) – Çevrilmiş birim sayısı
- **users** (*int*) – Kullanıcı sayısı
- **changes** (*int*) – Değişiklik sayısı
- **projects** (*int*) – Proje sayısı
- **components** (*int*) – Bileşen sayısı
- **translations** (*int*) – Çeviri sayısı
- **languages** (*int*) – Kullanılan dil sayısı
- **checks** (*int*) – Tetiklenmiş kalite denetimleri sayısı
- **configuration_errors** (*int*) – Yapılandırma hataları sayısı
- **suggestions** (*int*) – Bekleyen öneri sayısı
- **celery_queues** (*object*) – Celery kuyruklarının uzunlukları. Bilgi almak için: *Celery ile arka plan görevlerini kullanmak*
- **name** (*string*) – Yapılandırılmış sunucu adı

1.12.19 Bildirim kancaları

Bildirim kancaları, dış uygulamaların sürüm denetimi sistemi deposunun güncellendiğini Weblate tarafına bildirmesini sağlar.

Depoları ayrı ayrı güncellemek için proje, bileşen ve çeviri depo uç noktalarını kullanabilirsiniz. Bilgi almak için `POST /api/projects/(string:project)/repository/`.

GET /hooks/update/(string: project) /
string: component/

2.6 sürümünden beri kullanım dışı: Erişim denetimi listesi ile kısıtlanmış projeler için kimlik doğrulaması ile düzgün çalışan `POST /api/components/(string:project)/(string:component)/repository/` kullanın.

Bir bileşenin güncellenmesini tetikler (sürüm denetimi sisteminden çekme ve çeviri değişikliklerini tarama).

GET /hooks/update/(string: project) /

2.6 sürümünden beri kullanım dışı: Erişim denetimi listesi ile kısıtlanmış projeler için kimlik doğrulaması ile düzgün çalışan `POST /api/projects/(string:project)/repository/` kullanın.

Bir projedeki tüm bileşenlerin güncellenmesini tetikler (sürüm denetimi sisteminden çekme ve çeviri değişikliklerini tarama).

POST /hooks/github/

GitHub bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Not: GitHub, doğrudan Weblate bilgilendirmelerini destekler: Depo ayarlarından Weblate hizmet kancasını etkinleştirin ve adres olarak Weblate kurulumunuzun adresini yazın.

Ayrıca bakınız:

GitHub değişikliklerini kendiliğinden almak

GitHub bütünleştirmesini ayarlama yönergeleri için

<https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks>

GitHub web kancaları hakkında genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/gitlab/

GitLab bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

GitLab değişikliklerini kendiliğinden almak

GitLab bütünleştirmesini ayarlama yönergeleri için

<https://docs.gitlab.com/ee/user/project/integrations/webhooks.html>

GitLab web kancaları hakkında genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/bitbucket/

Bitbucket bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Bitbucket değişikliklerini kendiliğinden almak

Bitbucket bütünleştirmesini ayarlama yönergeleri için

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/>

Bitbucket web kancaları hakkında genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/pagure/

3.3 sürümünde geldi.

Pagure bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Pagure değişikliklerini kendiliğinden almak

Pagure bütünleştirmesini ayarlama yönergeleri için

https://docs.pagure.org/pagure/usage/using_webhooks.html

Pagure web kancaları hakkında genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/azure/

3.8 sürümünde geldi.

Azure DevOps bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Not: Lütfen *Gönderilecek kaynak ayrıntıları* değerinin *Tümü* olarak ayarlandığından emin olun. Yoksa Weblate ile Azure deponuz eşleşemez.

Ayrıca bakınız:

Azure Repos değişikliklerini kendiliğinden almak

Azure bütünleştirmesini ayarlama yönergeleri için

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)

Azure DevOps web kancaları hakkında genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/gitea/

3.9 sürümünde geldi.

Gitea web kancası bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Gitea depoları değişikliklerini kendiliğinden almak

Gitea bütünleştirmesini ayarlama yönergeleri için

<https://docs.gitea.io/en-us/webhooks/>

Gitea web kancaları hakkında genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

POST /hooks/gitee/

3.9 sürümünde geldi.

Gitee web kancası bildirimlerini işleyen ve eşleşen bileşenleri kendiliğinden güncelleyen özel kanca.

Ayrıca bakınız:

Gitee depoları değişikliklerini kendiliğinden almak

Gitee bütünleştirmesini ayarlama yönergeleri için

<https://gitee.com/help/categories/40>

Gitee web kancaları hakkında genel bilgiler

ENABLE_HOOKS

Kancaları Weblate genelinde etkinleştirmek için

1.12.20 Dışa aktarmak

Weblate, verileri farklı şekillerde işlemenizi sağlamak için çeşitli dışa aktarma seçenekleri sunar.

GET /exports/stats/(string: project) /
string: component/

Sorgu parametreleri

- **format** (*string*) – Çıktı biçimi: json ya da csv

2.6 sürümünden beri kullanım dışı: Lütfen bunun yerine `GET /api/components/(string:project)/(string:component)/statistics/` ve `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/` kullanın. Erişim denetim listesi kullanılan projelere de erişilmesini sağlar.

Belirli bir bileşenin istatistiklerini belirtilen biçimde alır.

Örnek istek:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Örnek yanıt:

```
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/"
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

},
{
  "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/"
},
{
  "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 akışları

Çevirilerdeki değişiklikler RSS akışlarında dışa aktarılır.

GET /exports/rss/ (string: project) /
string: component/string: language/

Bir çevirideki son değişikliklerin RSS akışını alır.

GET /exports/rss/ (string: project) /
string: component/

Bir bileşendeki son değişikliklerin RSS akışını alır.

GET /exports/rss/ (string: project) /

Bir projedeki son değişikliklerin RSS bildirimini alır.

GET /exports/rss/language/ (string: language) /

Bir dildeki son değişikliklerin RSS akışını alır.

GET /exports/rss/

Weblate kopyasındaki son değişikliklerin RSS akışını alır.

Ayrıca bakınız:

[Vikipedi üzerinde RSS](#)

1.13 Weblate istemcisi

2.7 sürümünde geldi: 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 Kurulum

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 Docker kullanımı

The Weblate Client is also available as a Docker image.

The image is published on Docker Hub: <https://hub.docker.com/r/weblate/wlc>

Kurulum:

```
docker pull weblate/wlc
```

The Docker container uses Weblate's default settings and connects to the API deployed in localhost. The API URL and API_KEY can be configured through the arguments accepted by Weblate.

The command to launch the container uses the following syntax:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Örnek:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

You might want to pass your *Yapılandırma dosyaları* 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 Başlarken

The wlc configuration is stored in `~/.config/weblate` (see *Yapılandırma dosyaları* 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
```

Ayrıca bakınız:

Yapılandırma dosyaları

1.13.4 Özet

```
wlc [arguments] <command> [options]
```

Komutlar aslında hangi işlemin yapılması gerektiğini belirtir.

1.13.5 Açıklama

Weblate Client, Weblate yönetiminin *Weblate REST API uygulaması* üzerinden yapılmasını sağlayan bir Python kütüphanesi ve komut satırı uygulamasıdır. Komut satırı uygulaması **wlc** komutu ile çağrılabilir ve *wlc* modülü üzerinde bulunur.

Argümanlar

Çıktı biçimini veya hangi Weblate kopyasının kullanılacağını tanımlamak için aşağıdaki bağımsız değişkenler kullanılabilir. Bunlar herhangi bir komuttan önce yazılmalıdır.

--format {csv,json,text,html}

Çıktı biçimini belirtin.

--url URL

API adresini belirtir. Yapılandırma dosyasındaki tüm değerlerin yerini alır. Bilgi almak için: *Yapılandırma dosyaları*. Adres /api/ ile bitmelidir. Örnek: `https://hosted.weblate.org/api/`.

--key KEY

Kullanılacak API kullanıcı anahtarını belirtir. Yapılandırma dosyasındaki tüm değerlerin yerini alır. Bilgi almak için: *Yapılandırma dosyaları*. Anahtarınızı Weblate profilinizde bulabilirsiniz.

--config PATH

Yapılandırma dosyası yolunu değiştirir. Bilgi almak için: *Yapılandırma dosyaları*.

--config-section SECTION

Kullanımdaki yapılandırma dosyası bölümünü değiştirir. Bilgi almak için: *Yapılandırma dosyaları*.

Komutlar

Kullanılabilecek komutlar şunlardır:

version

Geçerli sürümü yazdırır.

list-languages

Weblate üzerinde kullanılan dilleri listeler.

list-projects

Weblate üzerindeki projeleri listeler.

list-components

Weblate üzerindeki bileşenleri listeler.

list-translations

Weblate üzerindeki çevirileri listeler.

show

Weblate nesnesini (çeviri, bileşen ya da proje) görüntüler.

ls

Weblate nesnesini (çeviri, bileşen ya da proje) listeler.

commit

Bir Weblate nesnesinde (çeviri, bileşen ya da proje) yapılan değişiklikleri işler.

pull

Uzak depo değişikliklerini Weblate nesnesine (çeviri, bileşen ya da proje) çeker.

push

Weblate nesnesindeki (çeviri, bileşen ya da proje) değişiklikleri uzak depoya gönderir.

reset

0.7 sürümünde geldi: wlc 0.7 sürümünden başlayarak destekleniyor.

Uzak depoya eşleşmesi için Weblate nesnesindeki (çeviri, bileşen ya da proje) değişiklikleri sıfırlar.

cleanup

0.9 sürümünde geldi: wlc 0.9 sürümünden başlayarak destekleniyor.

Uzak depoya eşleşmesi için bir Weblate nesnesindeki (çeviri, bileşen ya da proje) izlenmeyen değişiklikleri siler.

repo

Belirli bir Weblate nesnesi (çeviri, bileşen ya da proje) için depo durumunu görüntüler.

stats

Belirli bir Weblate nesnesi (çeviri, bileşen ya da proje) için ayrıntılı istatistikleri görüntüler.

lock-status

0.5 sürümünde geldi: wlc 0.5 sürümünden başlayarak destekleniyor.

Kilit durumunu görüntüler.

lock

0.5 sürümünde geldi: wlc 0.5 sürümünden başlayarak destekleniyor.

Weblate bileşenini kilitleyerek üzerinde daha fazla çeviri yapılmasını engeller.

unlock

0.5 sürümünde geldi: wlc 0.5 sürümünden başlayarak destekleniyor.

Weblate bileşeninin kilidini açarak çeviri yapılmasını sağlar.

changes

0.7 sürümünde geldi: wlc 0.7 ve Weblate 2.10 sürümünden başlayarak destekleniyor.

Belirli bir nesnenin değişikliklerini görüntüler.

download

0.7 sürümünde geldi: wlc 0.7 sürümünden başlayarak destekleniyor.

Bir çeviri dosyasını indirir.

--convert

Dosya biçimini dönüştürür. Belirtilmemişse sunucuda herhangi bir dönüştürme yapılmaz ve dosya depodan olduğu gibi indirilir.

--output

Çıktının kaydedileceği dosyayı belirtir. Belirtilmezse stdout çıkışına yazdırılır.

upload

0.9 sürümünde geldi: wlc 0.9 sürümünden başlayarak destekleniyor.

Bir çeviri dosyasını yükler.

--overwrite

Yükleme yapılırken var olan çevirilerin üzerine yazar.

--input

İçeriği okunacak dosya. Belirtilmemişse stdin girişinden okunur.

--method

Kullanılacak yükleme yöntemi. Bilgi almak için: *Import methods*.

--fuzzy

Belirsiz (düzenlenecek olarak imlenmiş) dizgeleri işleme (*empty*, *process*, *approve*)

--author-name

Geçerli kullanıcı yerine kullanılacak yazar adı

--author-email

Geçerli kullanıcı yerine kullanılacak yazar e-posta adresi

İpucu: Komutların kullanımı hakkında ayrıntılı bilgi almak için `--help` seçeneğini kullanabilirsiniz. Örneğin: `wlc ls --help`.

1.13.6 Yapılandırma dosyaları

.weblate, .weblate.ini, weblate.ini

1.6 sürümünde değişti: *.ini* uzantılı dosyalar da kabul edilir.

Projeye özel yapılandırma dosyası

C:\Users\NAME\AppData\weblate.ini

1.6 sürümünde geldi.

Windows üzerinde kullanıcı yapılandırma dosyası.

~/.config/weblate

Kullanıcı yapılandırma dosyası

/etc/xdg/weblate

Sistem geneli yapılandırma dosyası

Program XDG teknik özelliklerine uyar. Böylece `XDG_CONFIG_HOME` ya da `XDG_CONFIG_DIRS` ortam değişkenlerine göre yapılandırma dosyalarının konumunu ayarlayabilirsiniz. Windows üzerinde yapılandırma dosyası için yeğlenen konum APPDATA klasörüdür.

[weblate] bölümünde şu ayarlar yapılandırılabilir (bunu `--config-section` ile özelleştirebilirsiniz):

key

Weblate erişimi için kullanılacak API anahtarı.

url

API sunucusunun adresi. Varsayılan değer: `http://127.0.0.1:8000/api/`.

translation

Varsayılan çeviri yolu. Bileşen ya da proje.

Yapılandırma dosyası bir INI dosyasıdır. Örneğin:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Ek API anahtarları [keys] bölümüne kaydedilebilir:

```
[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Bu seçenek, sürüm denetimi sistemi deposundaki `.weblate` yapılandırmasını kullanırken anahtarları kişisel ayarlarınızda tutmanızı sağlar. Böylece `wlc` hangi sunucuyla konuşması gerektiğini bilir.

1.13.7 Örnekler

Geçerli program sürümünü yazdır:

```
$ wlc version
version: 0.1
```

Tüm projeleri listele:

```
$ 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/
```

Çeviri dosyasını yükle:

```
$ wlc upload project/component/language --input /tmp/hello.po
```

Ayrıca `wlc` hangi proje üzerinde çalışacak onu da belirleyebilirsiniz:

```
$ 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/
```

Bu kurulumla, geçerli projede bekleyen değişiklikler kolayca işlenebilir:

```
$ wlc commit
```

1.14 Weblate Python API

1.14.1 Kurulum

The Python API is shipped separately, you need to install the *Weblate istemcisi* (wlc) to have it.

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception `wlc.WeblateException`

Tüm özel durumlar için temel sınıf.

Weblate

class `wlc.Weblate` (*key*="", *url*=None, *config*=None)

Parametreler

- **key** (*str*) – Kullanıcı anahtarı
- **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*)

Parametreler

path (*str*) – Request path

Dönüş türü

object

Performs a single API GET call.

post (*path*, ***kwargs*)

Parametreler

path (*str*) – Request path

Dönüş türü

object

Performs a single API GET call.

1.14.3 wlc.config

WeblateConfig

class wlc.config.**WeblateConfig** (*section* = 'wlc')

Parametreler

section (*str*) – Configuration section to use

Configuration file parser following XDG specification.

load (*path* = *None*)

Parametreler

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*)

Parametreler

- **settings** (*list*) – Settings to override as list of tuples
- **stdout** (*object*) – stdout file object for printing output, uses `sys.stdout` as default
- **args** (*list*) – Command-line arguments to process, uses `sys.args` as default

Main entry point for command-line interface.

@wlc.main.**register_command** (*command*)

Decorator to register *Command* class in main parser used by *main()*.

Command

class wlc.main.**Command** (*args*, *config*, *stdout* = *None*)

Main class for invoking commands.

2.1 Yapılandırma yönergeleri

2.1.1 Weblate kurulumu

Docker ile kurmak

Docker için hazırlanmış Weblate dağıtımını kullanarak kişisel Weblate bulut sunucunuzu saniyeler içinde çalışır duruma getirebilirsiniz. Tüm Weblate bağımlılıkları zaten katılmıştır. Varsayılan veri tabanı olarak PostgreSQL kurulmuştur.

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve web sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Aşağıdaki örneklerde, `docker-compose` kurulumu yapılmış hazır bir Docker ortamınız olduğu varsayılmaktadır. Yönergeler için Docker belgelerine bakabilirsiniz.

1. `weblate-docker` deposunu kopyalayın:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Ayarlarınızı yaptığınız bir `docker-compose.override.yml` dosyası oluşturun. Ortam değişkenlerinin tam listesi için *Docker ortam değişkenleri* bölümüne bakabilirsiniz.

```
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
```

Not: `WEBLATE_ADMIN_PASSWORD` değişkeni ayarlanmamışsa, yönetici kullanıcı başta belirtilen rastgele bir parolayla oluşturulur.

Verilen örnekte, Weblate 80 numaralı bağlantı noktasını dinleyecek şekilde ayarlanmıştır. `docker-compose.override.yml` dosyasını düzenleyerek bağlantı noktasını değiştirebilirsiniz.

3. Weblate kapsayıcılarını başlatın:

```
docker-compose up
```

Weblate dağıtımınızın tadını çıkarın. `weblate` kapsayıcısına 80 numaralı bağlantı noktasından erişilebilir.

2.15-2 sürümünde değişti: Kurulum bir süre önce değiştirildi. Daha önce ayrı bir web sunucusu kapsayıcısı vardı.

2.15-2 sürümünden sonra web sunucusu Weblate kapsayıcısına eklendi.

3.7.1-6 sürümünde değişti: 2019 Temmuz ayında (3.7.1-6 sürümünden başlayarak), kapsayıcılar kök kullanıcı olarak çalışmayacak biçimde değiştirildi. Açık bağlantı noktası 80 yerine 8080 olarak değiştirildi.

Ayrıca bakınız:

Invoking management commands

Docker hub etiketini seçmek

Docker hub üzerinde şu etiketleri kullanabilirsiniz. Kullanılabilecek etiketlerin tam listesi için <https://hub.docker.com/r/weblate/weblate/tags/> adresine bakabilirsiniz.

Etiket adı	Açıklama	Kullanım amacı
latest	Son etiketlenmiş sürümle eşleşen kararlı Weblate sürümü	Üretim ortamında güncellemeleri yapmak
<VERSION>-<I	Weblate kararlı sürümü	Üretim ortamında iyi tanımlanmış dağıtım
edge	Docker kapsayıcısındaki geliştirme değişiklikleriyle Weblate kararlı sürümü (güncelleştirilmiş bağımlılıklar gibi)	Deneme ortamında güncellemeleri yapmak
edge-<DATE>-	Docker kapsayıcısındaki geliştirme değişiklikleriyle Weblate kararlı sürümü (güncelleştirilmiş bağımlılıklar gibi)	Hazırlama ortamında iyi tanımlanmış dağıtım
bleeding	Git üzerinden Weblate geliştirme sürümü	Yakında eklenecek Weblate özelliklerini denemek için güncelleme ekleri
bleeding-<D	Git üzerinden Weblate geliştirme sürümü	Yaklaşan Weblate özelliklerini denemek için iyi tanımlanmış dağıtım

Her kalıp yayınlanmadan önce denenir. Bu nedenle *bleeding* sürümünün kullanımı bile oldukça güvenli olmalıdır.

HTTPS desteği ile Docker kapsayıcısı

Genel dağıtım yönergeleri için [Kurulum](#) bölümüne bakabilirsiniz. Bu bölümde yalnızca oradakinden farklı olan durumlar ele alınmıştır.

Kendi SSL sertifikalarınızı kullanmak

3.8-3 sürümünde geldi.

Kullanmak istediğiniz bir SSL sertifikanız varsa, dosyaları Weblate veri birimine yerleştirmeniz yeterlidir (bilgi almak için: `:ref:docker-volume`):

- `ssl/fullchain.pem` dosyasında gerekli sertifika kuruluşunun sertifikaları ile birlikte kullanılacak sertifika bulunur
- `ssl/privkey.pem` dosyasında kişisel anahtar bulunur

Bu dosyaların ikisinin de sahibi, docker kapsayıcısını başlatan kullanıcı ile aynı olmalı ve dosya izinleri 600 olarak ayarlanmış olmalıdır (yalnızca sahibi olan kullanıcı tarafından okunabilir ve yazılabilir).

Ek olarak, Weblate kapsayıcısı SSL bağlantılarını artık 4443 numaralı bağlantı noktasından karşılar. Docker compose değişikliklerine HTTPS için bağlantı noktası yönlendirmeyi de katmak istersiniz:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

Aynı sunucuda başka siteler barındırıyorsanız, büyük olasılıkla 80 ve 443 numaralı bağlantı noktaları NGINX gibi bir ters vekil sunucu tarafından kullanılmaktadır. HTTPS bağlantısını NGINX üzerinden docker kapsayıcısına geçirmek için şu yapılandırmayı kullanabilirsiniz:

```
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>;
    }
}
```

<SITE_URL>, <SITE> ve <EXPOSED_DOCKER_PORT> seçeneklerine ortamınızın gerçek değerlerini yazın.

Let's Encrypt ile SSL sertifikası sağlamak

Genel kurulumda kendiliğinden oluşturulan [Let's Encrypt](#) SSL sertifikalarını kullanmak istiyorsanız, ters bir HTTPS vekil sunucusu eklemeniz gerekir. Bunun için ek bir Docker kapsayıcısı olan [https-portal](#) kullanılır. Bu, `docker-compose-https.yml` dosyası kullanılarak yapılır. Ardından ayarlarınızla bir `docker-compose-https.override.yml` dosyası oluşturun:

```
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'
```

docker-compose komutunu her çağırdığınızda, her iki dosyayı da belirtmeniz ve ardından şunları yapmanız gerekir:

```
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
```

Docker kapsayıcısını yükseltmek

Genellikle yalnızca Weblate kapsayıcısını güncellemek ve PostgreSQL kapsayıcısını olduğu sürümde tutmak iyi bir fikirdir. Çünkü PostgreSQL yükseltmesi oldukça zorlu bir işlemdir ve çoğu durumda fazla bir yarar sağlamaz.

4.10-1 sürümünde değişti: Weblate 4.10-1 sürümünden bu yana, Docker kapsayıcısı PostgreSQL 10 ya da üzerindeki sürümlere gerek duyan Django 4.0 kullanır. Lütfen Weblate yükseltmesinden önce Docker sürümünü yükseltin. Bilgi almak için: [4.9 sürümünden 4.10 sürümüne yükseltmek](#) ve [PostgreSQL kapsayıcısını yükseltmek](#).

Bu işlemi, var olan `docker-compose` ögesine bağlı kalarak ve son kalıpları alıp yeniden başlatarak yapabilirsiniz:


```
# 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
```

Weblate veri tabanının başlangıçta kendiliğinden aktarılması ve el ile yapılacak ek işlemlere gerek kalmaması beklenir.

Not: Weblate ana sürümler arasında yükseltme işlemini desteklemez. Örneğin, 3.x serisini kullanıyor ve 4.x serisine yükseltmek istiyorsanız, önce son 4.0.x-y kalıbına (yazı hazırlandığında 4.0.4-5 sürümüydü) yükseltin. Bu kalıp önce aktarımı yapıp ardından yeni sürümlere yükseltmeyi sağlar.

`docker-compose` deposunu da güncellemek isteyebilirsiniz. Ancak çoğu durumda buna gerek olmaz. PostgreSQL sunucusunu yükseltmek için [PostgreSQL kapsayıcısını yükseltmek](#) sayfasına bakabilirsiniz.

PostgreSQL kapsayıcısını yükseltmek

PostgreSQL kapsayıcıları sürümler arasında kendiliğinden yükseltmeyi desteklemez. El ile yükseltmeniz gerekir. Aşağıdaki adımlarda, yükseltme seçeneklerinden biri gösterilmektedir.

Ayrıca bakınız:

<https://github.com/docker-library/postgres/issues/37>

1. Weblate kapsayıcısını durdurun:

```
docker-compose stop weblate cache
```

2. Veri tabanını yedekleyin:

```
docker-compose exec database pg_dumpall --clean --username weblate > backup.sql
```

3. Veri tabanı kapsayıcısını durdurun:

```
docker-compose stop database
```

4. PostgreSQL birimini kaldırın:

```
docker-compose rm -v database
docker volume remove weblate-docker_postgres-data
```

5. `docker-compose.yml` dosyasını yeni PostgreSQL sürümünü kullanarak şekilde ayarlayın.

6. Veri tabanı kapsayıcısını başlatın:

```
docker-compose up -d database
```

7. Veri tabanını yedekten geri yükleyin:

```
cat backup.sql | docker-compose exec -T database psql --username weblate --
↳ dbname postgres
```

8. (İsteğe bağlı) Weblate kullanıcısının parolasını güncelleyin. Parolaları depolama yöntemi değiştirildiği için PostgreSQL 14 ya da 15 sürümüne geçiş yaparken bu işlemi yapmanız gerekebilir:

```
docker-compose exec -T database psql --username weblate --dbname postgres -c
↳ "ALTER USER weblate WITH PASSWORD 'weblate'"
```

9. Kalan tüm kapsayıcıları başlatın:

```
docker-compose up -d
```

Yönetici olarak oturum açın

Kapsayıcı kurulumundan sonra, `WEBLATE_ADMIN_PASSWORD` içinde belirtilen parolayla ya da ayarlanmamışsa başlangıçta oluşturulan rastgele bir parolayla `admin` kullanıcısı ile oturum açabilirsiniz.

`admin` parolasını sıfırlamak için `WEBLATE_ADMIN_PASSWORD` seçeneğine yeni parolayı ayarlayın ve kapsayıcıyı yeniden başlatın.

Ayrıca bakınız:

`WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

İşlem sayısı ve bellek kullanımı

Hem uWSGI hem de Celery için çalışan işlem sayısı, işlemci sayısına göre kendiliğinden belirlenir. Bu seçim, çoğu bulut sanal makinesi için iyi çalışır. Bu makinelerde genellikle az sayıda işlemci ve iyi miktarda bellek bulunur.

Çok fazla işlemci çekirdeğiniz varsa ve bellek sorunlarından kurtulursanız, işlem sayısını azaltmayı deneyin:

```
environment:
  WEBLATE_WORKERS: 2
```

Ayrıca her işleyici kategorisi için ince ayar yapabilirsiniz:

```
environment:
  WEB_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Ayrıca bakınız:

`WEBLATE_WORKERS`, `CELERY_MAIN_OPTIONS`, `CELERY_NOTIFY_OPTIONS`, `CELERY_MEMORY_OPTIONS`, `CELERY_TRANSLATE_OPTIONS`, `CELERY_BACKUP_OPTIONS`, `CELERY_BEAT_OPTIONS`, `WEB_WORKERS`

Yatay ölçeklendirme

4.6 sürümünde geldi.

Hizmeti yatay olarak ölçeklemek için birden çok Weblate kapsayıcısı çalıştırabilirsiniz. `/app/data` biriminin tüm kapsayıcılar tarafından paylaşılması gerekir. Bunun için GlusterFS gibi küme dosya sistemlerinin kullanılması önerilir. `/app/cache` birimi her kapsayıcı için ayrı olmalıdır.

Her Weblate kapsayıcısı, `WEBLATE_SERVICE` ortam değişkenini kullanarak rolü tanımlar. Bazı hizmetlerin kümede yalnızca bir kez çalışması gerektiğinden ve hizmetlerin sıralaması önemli olduğundan, belgelere dikkatle uyun.

Örnek kurulumu `docker-compose` deposunda `docker-compose-split.yml` olarak bulabilirsiniz.

Docker ortam değişkenleri

Weblate *Yapılandırma* öğelerinin çoğu, Docker kapsayıcısında aşağıda açıklanan ortam değişkenleri ile ayarlanabilir. Docker ortam değişkenlerinde bulunmayan bir ayar yapmanız gerekiyorsa bilgi almak için *Ortam değişkenlerinin yapılandırması*.

Genel ayarlar

WEBLATE_DEBUG

DEBUG kullanarak Django hata ayıklama kipini yapılandırır.

Örnek:

```
environment:
  WEBLATE_DEBUG: 1
```

Ayrıca bakınız:

Hata ayıklama kipini kapatın

WEBLATE_LOGLEVEL

Günlük kayıtlarının ayrıntı düzeyini yapılandırır.

WEBLATE_LOGLEVEL_DATABASE

Veri tabanı sorguları günlük kayıtlarının ayrıntı düzeyini yapılandırır.

WEBLATE_SITE_TITLE

Tüm sayfaların üst bilgisi olarak görüntülenecek site başlığını değiştirir.

WEBLATE_SITE_DOMAIN

Site etki alanını yapılandırır. Bu parametre gereklidir.

Ayrıca bakınız:

*Doğru site etki alanını ayarlayın, *SITE_DOMAIN**

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Site yöneticisinin adını ve e-posta adresini yapılandırır. Hem *ADMINS* ayarını hem de *admin* kullanıcıyı oluşturmak için kullanılır (bilgi almak için: *WEBLATE_ADMIN_PASSWORD*).

Örnek:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

Ayrıca bakınız:

*Yönetici olarak oturum açın, Yöneticileri düzgün şekilde yapılandırın, *ADMINS**

WEBLATE_ADMIN_PASSWORD

admin kullanıcısının parolasını ayarlar.

- Ayarlanmamışsa ve *admin* kullanıcısı yoksa, kapsayıcı ilk kez başlatıldığında görüntülenen rastgele bir parola ile oluşturulur.
- Ayarlanmamışsa ve *admin* kullanıcısı varsa, hiçbir işlem gerçekleştirilmez.
- Ayarlanmışsa, *admin* kullanıcısı her kapsayıcı başlangıcında *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_NAME* ve *WEBLATE_ADMIN_EMAIL* seçeneklerine göre ayarlanır.

Uyarı: Parolayı yapılandırma dosyasında depolamak bir güvenlik riski oluşturabilir. Bu değişkeni yalnızca ilk kurulum için (ya da Weblate ilk kez başlatıldığında rastgele parola oluşturmaya izin verin) ya da parola kurtarma için kullanmayı değerlendirin.

Ayrıca bakınız:

Yönetici olarak oturum açın, `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_PASSWORD_FILE`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

WEBLATE_ADMIN_PASSWORD_FILE

Admin kullanıcısının parolasının bulunduğu bir dosyanın yolunu ayarlar.

Ayrıca bakınız:

`WEBLATE_ADMIN_PASSWORD`

WEBLATE_SERVER_EMAIL

Hata iletilerinin gönderileceği e-posta adresi.

Ayrıca bakınız:

`SERVER_EMAIL`, [E-posta gönderimini yapılandırın](#)

WEBLATE_DEFAULT_FROM_EMAIL

Giden e-postaların adresini yapılandırır.

Ayrıca bakınız:

`DEFAULT_FROM_EMAIL`, [E-posta gönderimini yapılandırın](#)

WEBLATE_CONTACT_FORM

İletişim formu davranışını yapılandırır. Bilgi almak için: [CONTACT_FORM](#).

WEBLATE_ALLOWED_HOSTS

`ALLOWED_HOSTS` kullanarak izin verilen HTTP sunucu adlarını yapılandırır.

Varsayılan değer: *. Tüm sunucu adlarına izin verir.

Örnek:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com,example.com
```

Ayrıca bakınız:

`ALLOWED_HOSTS`, [Allowed hosts kurulumu](#), [Doğru site etki alanını ayarlayın](#)

WEBLATE_REGISTRATION_OPEN

`REGISTRATION_OPEN` seçeneğini değiştirerek hesap açma özelliğini etkinleştirir.

Örnek:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

`REGISTRATION_ALLOW_BACKENDS` seçeneği ile yeni hesap açmak için hangi kimlik doğrulama yöntemlerinin kullanılabileceğini yapılandırır.

Örnek:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
  ↪oauth2
```

WEBLATE_REGISTRATION_REBIND

4.16 sürümünde geldi.

`REGISTRATION_REBIND` seçeneğini yapılandırır.

WEBLATE_TIME_ZONE

Weblate üzerinde kullanılacak saat dilimini yapılandırır. Bilgi almak için: `TIME_ZONE`.

Not: Docker kapsayıcısının saat dilimini değiştirmek için TZ ortam değişkenini kullanın.

Örnek:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Weblate uygulamasının ters HTTPS vekil sunucu arkasında çalıştığını varsaymasını sağlar. Weblate e-posta ve API bağlantılarında HTTPS kullanılmasını sağlar ya da çerezler için güvenli imlerini ayarlar.

İpucu: Olası uyarılar için `ENABLE_HTTPS` belgelerine bakabilirsiniz.

Not: Bu seçenek, Weblate kapsayıcısının HTTPS bağlantılarını kabul etmesini sağlamaz. Bunu da yapılandırmanız gerekir. Bilgi almak için: *HTTPS desteği ile Docker kapsayıcısı*.

Örnek:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Ayrıca bakınız:

`ENABLE_HTTPS` Doğru site etki alanını ayarlayın, `WEBLATE_SECURE_PROXY_SSL_HEADER`

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

4.12.1 sürümünde geldi.

Weblate belgelerinin başındaki `meta[name=monetization]` alanının ayarlanmasını sağlar. Birden çok seçenek belirtilmişse, rastgele olarak biri seçilir.

Ayrıca bakınız:

`INTERLEDGER_PAYMENT_POINTERS`

WEBLATE_IP_PROXY_HEADER

IP adresinin Weblate tarafından herhangi bir HTTP üst bilgisinden alınmasını sağlar. Weblate kapsayıcısının önünde ters vekil sunucu varsa bunu kullanın.

`IP_BEHIND_REVERSE_PROXY` özelliğini etkinleştirir ve `IP_PROXY_HEADER` ögesini ayarlar.

Not: Biçim, Django beklentilerine uygun olmalıdır. Django ham HTTP üst bilgi adlarını şu biçimde **dönüştürür**:

- tüm karakterleri büyük harfe dönüştürür
- tire karakterini alt çizgi karakterine dönüştürür
- başa HTTP_ ön ekini ekler

Böylece X-Forwarded-For değeri HTTP_X_FORWARDED_FOR ile eşleştirilir.

Örnek:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Bir isteği belirten HTTP üst bilgi/değer ikilisini temsil eden bir demet güvenlidir. Weblate, standart HTTPS üst bilgilerini geçirmeyen SSL sonlandırma işlemi yapan bir ters vekil sunucu arkasında ise bu gereklidir.

Örnek:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

Ayrıca bakınız:

SECURE_PROXY_SSL_HEADER

WEBLATE_REQUIRE_LOGIN

Tüm Weblate üzerinde kimlik doğrulamasını zorunlu kılmak için *REQUIRE_LOGIN* seçeneğini etkinleştirir.

Örnek:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

LOGIN_REQUIRED_URLS_EXCEPTIONS seçeneğini kullanarak tüm Weblate kurulumu için gereken kimlik doğrulama adresi istisnalarını ekler.

Tüm ayarları değiştirebilir ya da ADD ve REMOVE seçeneklerini kullanarak varsayılan değeri değiştirebilirsiniz.

WEBLATE_GOOGLE_ANALYTICS_ID

GOOGLE_ANALYTICS_ID seçeneğini değiştirerek Google Analytics kimliğini yapılandırır.

WEBLATE_GITHUB_USERNAME

WEBLATE_GITHUB_TOKEN

WEBLATE_GITHUB_HOST

GITHUB_CREDENTIALS seçeneğini değiştirerek GitHub çekme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

GitHub çekme istekleri

WEBLATE_GITLAB_USERNAME

WEBLATE_GITLAB_TOKEN

WEBLATE_GITLAB_HOST

GITLAB_CREDENTIALS seçeneğini değiştirerek GitLab birleştirme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

GitLab birleştirme istekleri

WEBLATE_GITEA_USERNAME

WEBLATE_GITEA_TOKEN**WEBLATE_GITEA_HOST**

GITEA_CREDENTIALS seçeneğini değiştirerek Gitea çekme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

Gitea çekme isteği

WEBLATE_PAGURE_USERNAME**WEBLATE_PAGURE_TOKEN****WEBLATE_PAGURE_HOST**

PAGURE_CREDENTIALS seçeneğini değiştirerek Pagure birleştirme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

Pagure birleştirme istekleri

WEBLATE_BITBUCKETSERVER_USERNAME**WEBLATE_BITBUCKETSERVER_TOKEN****WEBLATE_BITBUCKETSERVER_HOST**

BITBUCKETSERVER_CREDENTIALS seçeneğini değiştirerek Bitbucket sunucusu birleştirme istekleri bütünleştirmesini yapılandırır.

Ayrıca bakınız:

Bitbucket sunucusu çekme isteği

WEBLATE_DEFAULT_PULL_MESSAGE

DEFAULT_PULL_MESSAGE seçeneğini değiştirerek API ile yapılan çekme isteklerinin varsayılan başlığını ve iletiyi yapılandırır

Ayrıca bakınız:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

Dil basitleştirme ilkesini yapılandırır. Bilgi almak için: `:setting: SIMPLIFY_LANGUAGES`.

WEBLATE_DEFAULT_ACCESS_CONTROL

Yeni projeler için varsayılan *Erişim denetimi* ögesini yapılandırır. Bilgi almak için: *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Yeni bileşenler için *Kısıtlı erişim* seçeneğinin varsayılan değerini yapılandırır. Bilgi almak için: *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Yeni bileşenler için *Çevirilerin yayılmasını sağlamak* seçeneğinin varsayılan değerini yapılandırır. Bilgi almak için: *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

DEFAULT_COMMITER_EMAIL seçeneğini yapılandırır.

WEBLATE_DEFAULT_COMMITER_NAME

DEFAULT_COMMITER_NAME seçeneğini yapılandırır.

WEBLATE_DEFAULT_SHARED_TM

DEFAULT_SHARED_TM seçeneğini yapılandırır.

WEBLATE_AKISMET_API_KEY

Akismet API anahtarını yapılandırır. Bilgi almak için: *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Gönderimlerin GPG imzalamasını yapılandırır. Bilgi almak için: [WEBLATE_GPG_IDENTITY](#).

Ayrıca bakınız:

Git işlemlerini GnuPG ile imzalamak

WEBLATE_URL_PREFIX

Weblate ortamının çalıştığı adres ön ekini yapılandırır. Bilgi almak için: [URL_PREFIX](#).

WEBLATE_SILENCED_SYSTEM_CHECKS

Görüntülenmesini istemediğiniz denetimleri yapılandırır. Bilgi almak için: [SILENCED_SYSTEM_CHECKS](#).

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Content-Security-Policy HTTP üst bilgisinin özelleştirilmesini sağlar.

Ayrıca bakınız:

İçerik güvenliği ilkesi, CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

WEBLATE_LICENSE_FILTER

[LICENSE_FILTER](#) seçeneğini yapılandırır.

WEBLATE_LICENSE_REQUIRED

[LICENSE_REQUIRED](#) seçeneğini yapılandırır

WEBLATE_WEBSITE_REQUIRED

[WEBSITE_REQUIRED](#) seçeneğini yapılandırır

WEBLATE_HIDE_VERSION

[HIDE_VERSION](#) seçeneğini yapılandırır.

WEBLATE_BASIC_LANGUAGES

[BASIC_LANGUAGES](#) seçeneğini yapılandırır.

WEBLATE_DEFAULT_AUTO_WATCH

[DEFAULT_AUTO_WATCH](#) seçeneğini yapılandırır.

WEBLATE_RATELIMIT_ATTEMPTS

WEBLATE_RATELIMIT_LOCKOUT

WEBLATE_RATELIMIT_WINDOW

4.6 sürümünde geldi.

Hız sınırlayıcıyı yapılandırır.

İpucu: Yapılandırmayı herhangi bir hız sınırlayıcı kapsamı için ayarlayabilirsiniz. Bunu yapmak için, `:ref:rate-limit``` içinde belirtilen ayarlardan herhangi birine ``WEBLATE_ ön ekini ekleyin.

Ayrıca bakınız:

Hız sınırlama, RATELIMIT_ATTEMPTS, RATELIMIT_WINDOW, RATELIMIT_LOCKOUT

WEBLATE_API_RATELIMIT_ANON**WEBLATE_API_RATELIMIT_USER**

4.11 sürümünde geldi.

API hız sınırlamasını yapılandırır. Varsayılan değer anonim kullanıcılar için 100/gün ve kimliği doğrulanmış kullanıcılar için ``5000/saat` biçimindedir.

Ayrıca bakınız:

API hız sınırlaması

WEBLATE_ENABLE_HOOKS

4.13 sürümünde geldi.

ENABLE_HOOKS seçeneğini yapılandırır.

WEBLATE_ENABLE_AVATARS

4.6.1 sürümünde geldi.

ENABLE_AVATARS seçeneğini yapılandırır.

WEBLATE_AVATAR_URL_PREFIX

4.15 sürümünde geldi.

AVATAR_URL_PREFIX seçeneğini yapılandırır.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

4.9 sürümünde geldi.

LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH seçeneğini yapılandırır.

WEBLATE_SSH_EXTRA_ARGS

4.9 sürümünde geldi.

SSH_EXTRA_ARGS seçeneğini yapılandırır.

WEBLATE_BORG_EXTRA_ARGS

4.9 sürümünde geldi.

BORG_EXTRA_ARGS seçeneğini yapılandırır.

WEBLATE_ENABLE_SHARING

4.14.1 sürümünde geldi.

ENABLE_SHARING seçeneğini yapılandırır.

WEBLATE_EXTRA_HTML_HEAD

4.15 sürümünde geldi.

EXTRA_HTML_HEAD seçeneğini yapılandırır.

WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE

4.15 sürümünde geldi.

PRIVATE_COMMIT_EMAIL_TEMPLATE seçeneğini yapılandırır.

WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN

4.15 sürümünde geldi.

PRIVATE_COMMIT_EMAIL_OPT_IN seçeneğini yapılandırır.

WEBLATE_CORS_ALLOWED_ORIGINS

4.16 sürümünde geldi.

Belirtilen kaynaklardan CORS istekleri yapılabilmesini sağlar.

Örnek:

```
environment:
  WEBLATE_CORS_ALLOWED_ORIGINS: https://example.com,https://weblate.org
```

Kendiliğinden öneri ayarları

4.13 sürümünde değişti: Kendiliğinden öneri hizmetleri artık kullanıcı arayüzünden yapılandırılıyor. Bilgi almak için: [Kendiliğinden önerileri yapılandırmak](#).

Var olan ortam değişkenleri Weblate 4.13 sürümüne geçiş yapılırken içe aktarılır. Ancak bunların değiştirilmesinin başkaca bir etkisi olmaz.

Kimlik doğrulama ayarları

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

LDAP sunucusunun bind parolasını içeren dosyanın yolu.

Ayrıca bakınız:

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

LDAP kimlik doğrulama yapılandırması.

Doğrudan bağlama örneği:

```
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
```

Arama ve bağlama örneği:

```
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
```

Birleşik arama ve bağlama örneği:

```
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
```

Active Directory üzerinde arama ve bağlama örneği:

```
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)
```

Ayrıca bakınız:

LDAP kimlik doğrulaması

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

GitHub kimlik doğrulaması seçeneğini etkinleştirir.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Bitbucket kimlik doğrulaması seçeneğini etkinleştirir.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Facebook OAuth 2 seçeneğini etkinleştirir.

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

Google OAuth 2 seçeneğini etkinleştirir.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

GitLab OAuth 2 seçeneğini etkinleştirir.

Gitea

WEBLATE_SOCIAL_AUTH_GITEA_API_URL

WEBLATE_SOCIAL_AUTH_GITEA_KEY

WEBLATE_SOCIAL_AUTH_GITEA_SECRET

Gitea kimlik doğrulaması kullanılsın.

Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET

Azure Active Directory kimlik doğrulamasını etkinleştirir. Bilgi almak için: [Microsoft Azure Active Directory](#).

Kiracı destekli Azure Active Directory

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID

Kiracı destekli Azure Active Directory kimlik doğrulamasını etkinleştirir. Bilgi almak için: [Microsoft Azure Active Directory](#).

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

Keycloak kimlik doğrulamasını etkinleştirir. Bilgi almak için: [Belgeler](#).

Linux hizmet sağlayıcıları

Şu değişkenleri bir değere ayarlayarak Linux üreticilerinin kimlik doğrulama hizmetleri ile kimlik doğrulamayı etkinleştirebilirsiniz.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Slack kimlik doğrulamasını etkinleştirir. Bilgi almak için: [Slack](#).

OpenID Connect

4.13-1 sürümünde geldi.

WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT

WEBLATE_SOCIAL_AUTH_OIDC_KEY

WEBLATE_SOCIAL_AUTH_OIDC_SECRET

WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY

Genel OpenID Connect bütünleştirmesini yapılandırır.

Ayrıca bakınız:

[OIDC \(OpenID Connect\)](#)

SAML

Kendiliğinden imzalanan SAML anahtarları, ilk kapsayıcı başlatıldığında kendiliğinden oluşturulur. Kendi anahtarlarınızı kullanmak isterseniz, sertifikayı ve kişisel anahtarı `/app/data/ssl/saml.crt` ve `/app/data/ssl/saml.key` dosyalarına yerleştirin.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

WEBLATE_SAML_IDP_IMAGE

WEBLATE_SAML_IDP_TITLE

SAML kimlik hizmeti sağlayıcısı ayarları. Bilgi almak için: [SAML kimlik doğrulaması](#).

Diğer kimlik doğrulama ayarları

WEBLATE_NO_EMAIL_AUTH

Herhangi bir değere ayarlandığında e-posta kimlik doğrulamasını devre dışı bırakır. Bilgi almak için: [Parola ile kimlik doğrulamayı kapatmak](#).

PostgreSQL veri tabanı kurulumu

Veri tabanı `docker-compose.yml` tarafından oluşturulur. Bu nedenle bu ayarlar hem Weblate hem de PostgreSQL kapsayıcılarını etkiler.

Ayrıca bakınız:

[Weblate için veri tabanı kurulumu](#)

POSTGRES_PASSWORD

PostgreSQL parolası.

POSTGRES_PASSWORD_FILE

PostgreSQL parolasının bulunduğu dosyanın yolu. POSTGRES_PASSWORD alternatif olarak kullanın.

POSTGRES_USER

PostgreSQL kullanıcı adı.

POSTGRES_DATABASE

PostgreSQL veri tabanı adı.

POSTGRES_HOST

PostgreSQL sunucusunun adı ya da IP adresi. Varsayılan değer: `database`.

POSTGRES_PORT

PostgreSQL sunucu bağlantı noktası. Varsayılan değer: Yok (varsayılan değeri kullanır).

POSTGRES_SSL_MODE

PostgreSQL sunucusunun SSL bağlantılarını nasıl işleyeceğini yapılandırın. Kullanılabilecek seçenekler için [SSL kipi açıklamaları](#) bölümüne bakabilirsiniz

POSTGRES_ALTER_ROLE

Aktarım sırasında değiştirilecek rol adını yapılandırır. Bilgi almak için: [Weblate yapılandırmasını PostgreSQL kullanacak biçimde ayarlamak](#).

POSTGRES_CONN_MAX_AGE

4.8.1 sürümünde geldi.

Saniye ve tamsayı olarak veri tabanı bağlantısının yaşam süresi. Her isteğin sonunda veri tabanı bağlantılarını kapatılması için 0 yazın (varsayılan davranış).

Bağlantı kalıcılığını etkinleştirmek genellikle veri tabanına daha açık bir bağlantı kurulmasına neden olur. Lütfen bu seçeneği etkinleştirmeden önce veri tabanı yapılandırmanızı ayarlayın.

Örnek yapılandırma:

```
environment:
  POSTGRES_CONN_MAX_AGE: 3600
```

Ayrıca bakınız:

`CONN_MAX_AGE`, [Persistent connections](#)

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

4.9.1 sürümünde geldi.

Veri tabanındaki sunucu tarafı imleçlerini devre dışı bırakın. Bu uygulama, bazı **pgbouncer** kurulumlarında gereklidir.

Örnek yapılandırma:

```
environment:
  POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Ayrıca bakınız:

`DISABLE_SERVER_SIDE_CURSORS`, [Transaction pooling and server-side cursors](#)

Veri tabanı yedekleme ayarları

Ayrıca bakınız:

Yedeklemeler için dökümü yapılan veriler

WEBLATE_DATABASE_BACKUP

`DATABASE_BACKUP` kullanarak günlük veri tabanı dökümünü yapılandırır. Varsayılan değer: `plain`.

Ön bellek sunucusu kurulumu

Weblate, Redis kullanılmasını önemle önerir ve Weblate Docker üzerinde çalıştırılırken bir Redis kopyası sağlamanız gerekir.

Ayrıca bakınız:

Ön bellek özelliğini açın

REDIS_HOST

Redis sunucusunun adı ya da IP adresi. Varsayılan değer: `cache`.

REDIS_PORT

Redis sunucusunun bağlantı noktası. Varsayılan değer: `6379`.

REDIS_DB

Redis veri tabanı numarası. Varsayılan değer: `1`.

REDIS_PASSWORD

Redis sunucusunun parolası. Varsayılan olarak kullanılmaz.

REDIS_PASSWORD_FILE

Redis sunucu parolasının bulunduğu dosyanın yolu.

Ayrıca bakınız:

`REDIS_PASSWORD`

REDIS_TLS

Redis bağlantısı için SSL kullanımını etkinleştirir.

REDIS_VERIFY_SSL

Redis bağlantısı için SSL sertifikası doğrulamasını devre dışı bırakmak için kullanılabilir.

E-posta sunucusu kurulumu

Giden e-postanın çalışmasını sağlamak için, bir e-posta sunucusu belirtmeniz gerekir.

Örnek TLS yapılandırması:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Örnek SSL yapılandırması:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_PORT: 465
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
WEBLATE_EMAIL_USE_TLS: 0
WEBLATE_EMAIL_USE_SSL: 1
```

Ayrıca bakınız:*Giden e-postayı yapılandırmak***WEBLATE_EMAIL_HOST**

E-posta sunucusunun adı ya da IP adresi.

Ayrıca bakınız:

```
WEBLATE_EMAIL_PORT, WEBLATE_EMAIL_USE_SSL, WEBLATE_EMAIL_USE_TLS, EMAIL_HOST
```

WEBLATE_EMAIL_PORT

E-posta sunucusunun bağlantı noktası. Varsayılan değer: 25.

Ayrıca bakınız:

```
EMAIL_PORT
```

WEBLATE_EMAIL_HOST_USER

E-posta ile kimlik doğrulama kullanıcısı.

Ayrıca bakınız:

```
EMAIL_HOST_USER
```

WEBLATE_EMAIL_HOST_PASSWORD

E-posta ile kimlik doğrulama parolası.

Ayrıca bakınız:

```
EMAIL_HOST_PASSWORD
```

WEBLATE_EMAIL_HOST_PASSWORD_FILE

E-posta ile kimlik doğrulama parolasını içeren dosyanın yolu.

Ayrıca bakınız:

```
WEBLATE_EMAIL_HOST_PASSWORD
```

WEBLATE_EMAIL_USE_SSL

SMTP sunucusuyla iletişim için örtülü TLS (güvenli) bağlantısının kullanılıp kullanılmayacağını yapılandırır. Çoğu e-posta belgesinde, bu tür TLS bağlantısı SSL olarak adlandırılır. Genellikle 465 numaralı bağlantı noktasıyla kullanılır. Sorun yaşıyorsanız [WEBLATE_EMAIL_USE_TLS](#) açık TLS ayarına bakın.

4.11 sürümünde değişti: SSL/TLS desteği [WEBLATE_EMAIL_PORT](#) seçeneğine göre kendiliğinden etkinleştirilir.

Ayrıca bakınız:

```
WEBLATE_EMAIL_PORT, WEBLATE_EMAIL_USE_TLS, EMAIL_USE_SSL
```

WEBLATE_EMAIL_USE_TLS

SMTP sunucusuyla iletişim için TLS (güvenli) bağlantısının kullanılıp kullanılmayacağını yapılandırır. Bu seçenek, genellikle 587 ya da 25 numaralı bağlantı noktalarında bulunan açık TLS bağlantıları için kullanılır. Bağlantıların askıda kalması sorunu yaşıyorsanız envvar: [WEBLATE_EMAIL_USE_SSL](#) seçeneğindeki örtülü TLS ayarına bakın.

4.11 sürümünde değişti: SSL/TLS desteği [WEBLATE_EMAIL_PORT](#) seçeneğine göre kendiliğinden etkinleştirilir.

Ayrıca bakınız:

```
WEBLATE_EMAIL_PORT, WEBLATE_EMAIL_USE_SSL, EMAIL_USE_TLS
```

WEBLATE_EMAIL_BACKEND

Django arka ucunu e-posta göndermek için yapılandırır.

Ayrıca bakınız:

E-posta gönderimini yapılandırın, `EMAIL_BACKEND`

WEBLATE_AUTO_UPDATE

Depoların Weblate tarafından güncellenip güncellenmeyeceğini ve nasıl güncelleneceğini yapılandırır.

Ayrıca bakınız:

AUTO_UPDATE

Not: Bu bir Boolean ayarıdır ("true" ya da "false" kullanın).

Site bütünleştirmesi

WEBLATE_GET_HELP_URL

GET_HELP_URL seçeneğini yapılandırır.

WEBLATE_STATUS_URL

STATUS_URL seçeneğini yapılandırır.

WEBLATE_LEGAL_URL

LEGAL_URL seçeneğini yapılandırır.

WEBLATE_PRIVACY_URL

PRIVACY_URL seçeneğini yapılandırır.

Hataları bildirmek

Kurulumdaki hataları sistematik olarak toplamamız önerilir. Bilgi almak için: *Hata raporlarını derlemek*.

Rollbar desteğini etkinleştirmek için şunları ayarlayın:

ROLLBAR_KEY

Rollbar ileti sunucunuzun erişim kodu.

ROLLBAR_ENVIRONMENT

Rollbar ortamınız. Varsayılan değer: `production`.

Sentry desteğini etkinleştirmek için şunları ayarlayın:

SENTRY_DSN

Sentry DSN değerinizi.

SENTRY_ENVIRONMENT

Sentry ortamınız (isteğe bağlı).

Yerelleştirme CDN

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

4.2.1 sürümünde geldi.

JavaScript yerelleştirme CDN yapılandırması.

WEBLATE_LOCALIZE_CDN_PATH kapsayıcı içindeki yoldur. Geçici depolama biriminde değil, kalıcı birimde olmalıdır.

Olasılıklardan biri, Weblate verilerinin içinde tutulmasıdır:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Not: Weblate tarafından oluşturulan dosyaların sunulmasını ayarlamak sizin sorumluluğunuzdur. Bu seçenek yalnızca dosyaları yapılandırılmış konumda tutar.

Ayrıca bakınız:

weblate-cdn, *LOCALIZE_CDN_URL*, *LOCALIZE_CDN_PATH*

Etkin uygulamaları, denetimleri, eklentileri ya da kendiliğinden düzeltmeleri değiştirmek

3.8-5 sürümünde geldi.

Etkin denetimlerin, eklentilerin ya da kendiliğinden düzeltmelerin iç yapılandırması şu değişkenlerle ayarlanabilir:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Örnek:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
  ↳ SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

Ayrıca bakınız:

CHECK_LIST, *AUTOFIX_LIST*, *WEBLATE_ADDONS*, *INSTALLED_APPS*

Kapsayıcı ayarları

WEBLATE_WORKERS

4.6.1 sürümünde geldi.

Kapsayıcıda çalışan işlemlerin temel sayısı. Ayarlanmadığında, kullanılabilir işlemci çekirdeği sayısına bağlı olarak kapsayıcı başlangıcında kendiliğinden belirlenir.

`CELERY_MAIN_OPTIONS`, `CELERY_NOTIFY_OPTIONS`, `CELERY_MEMORY_OPTIONS`, `CELERY_TRANSLATE_OPTIONS`, `CELERY_BACKUP_OPTIONS`, `CELERY_BEAT_OPTIONS`, ve `WEB_WORKERS` seçeneklerini ayarlamak için kullanılır. Bu seçenekleri ince ayar yapmak için kullanabilirsiniz.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

Bu seçenekler, Celery işlemi seçeneklerini ayarlamanızı sağlar. Eşzamanlılığı ayarlamak (`--concurrency 16`) veya farklı havuz uygulaması kullanmak (`--pool =gevent`) yararlı olabilir.

Varsayılan olarak, eşzamanlı işlem sayısı `WEBLATE_WORKERS` seçeneğinden alınır.

Örnek:

```
environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```

Ayrıca bakınız:

[Celery işlemi seçenekleri](#), [Celery ile arka plan görevlerini kullanmak](#)

WEB_WORKERS

Çalışacak uWSGI işlemi sayısını yapılandırın.

Varsayılan değer: `WEBLATE_WORKERS`.

Örnek:

```
environment:
  WEB_WORKERS: 32
```

WEBLATE_SERVICE

Kapsayıcı içinde hangi hizmetlerin yürütülmesi gerektiğini tanımlar. Bunu [Yatay ölçeklendirme](#) için kullanın.

Şu hizmetler tanımlanmıştır:

celery-beat

Celery görev zamanlayıcısı. Yalnızca bir kopya çalışıyor olmalıdır. Bu kapsayıcı aynı zamanda veri tabanı yapısı aktarımlarından da sorumludur ve diğerlerinden önce başlatılmalıdır.

celery-backup

Yedeklemeler için Celery işlemi. Yalnızca bir kopya çalışmalıdır.

celery-celery

Genel Celery işlemi.

celery-memory

Çeviri belleği Celery işlemi.

celery-notify

Bildirim Celery işlemi.

celery-translate

Kendiliğinden çeviri Celery işlemi.

web

Web sunucusu.

Docker kapsayıcısı birimleri

Weblate kapsayıcısı tarafından dışa aktarılan iki birim (veri ve ön bellek) bulunur. Ayrıca diğer hizmet kapsayıcılarının (PostgreSQL ya da Redis) da veri birimleri vardır. Ancak bunlar bu belgenin kapsamında değildir.

Veri birimi, kopyalanmış depolar gibi Weblate kalıcı verilerini depolamak veya Weblate kurulumunu özelleştirmek için kullanılır.

Docker biriminin barındırma sistemindeki konumu Docker yapılandırmanıza bağlıdır. Ancak genellikle `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/` (yol, docker-compose klasörünün adı, kapsayıcı ve birim adlarından oluşur) içinde tutulur. Kapsayıcıda `/app/data` olarak bağlanır.

Ön bellek birimi `/app/cache` olarak bağlanır ve durağan dosyalar ile `CACHE_DIR` klasörünü tutmak için kullanılır. İçeriği kapsayıcı başlangıcında yeniden oluşturulur ve birim `tmpfs` gibi kısa ömürlü dosya sistemi kullanılarak bağlanabilir.

Birimleri el ile oluştururken, klasörlerin sahibi kapsayıcı içindeki UID 1000 olmalıdır.

Ayrıca bakınız:

[Docker birimleri belgeleri](#)

Ortam değişkenlerinin yapılandırması

Docker ortam değişkenleri, Weblate kurulumları ile ilgili pek çok *yapılandırma ayarını* karşılamayı amaçlıyor.

Bir ortam değişkeni olarak karşılığı olmayan bir ayar bulursanız ve var olması gerektiğine inanıyorsanız, *Gelecek Weblate sürümlerine eklenmesini istemekten çekinmeyin*.

Docker ortam değişkeni olarak karşılığı olmayan bir ayarı değiştirmeniz gerekirse, bunu *veri biriminden* ya da *Docker kalıbını genişleterek* yapabilirsiniz.

Ayrıca bakınız:

Weblate platformunu özelleştirmek

Veri biriminden alınan ayarları değiştirmek

Ortam değişkenleri ile tanımlanan ayarları genişletmek ya da değiştirmek için veri birimi kök klasöründe bir `/app/data/settings-override.py` dosyası oluşturabilirsiniz.

Docker kalıbını genişleterek ayarları değiştirmek

Ayarları veri birimi yerine Docker kalıbı düzeyinde değiştirmek için:

1. *Özel bir Python paketi oluşturmak.*
2. Paketinize, `weblate.settings_docker` ögesinden tüm ayarları içe aktaran bir modül ekleyin.

Örneğin, *Bir Python modülü oluşturmak* adresinde tanımlanan örnek paket yapısı içinde, şu başlangıç koduyla bir `weblate_customization/weblate_customization/settings.py` dosyası oluşturabilirsiniz:

```
from weblate.settings_docker import *
```

3. Resmi Weblate Docker kalıbından devralınan ve ardından paketinizi kuran ve ayarlar modülünüze `DJANGO_SETTINGS_MODULE` ortam değişkenini gösteren özel bir `Dockerfile` oluşturun:

```
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. Resmi Weblate Docker kalıbını kullanmak yerine, bu `Dockerfile` dosyasından özel bir kalıp oluşturun.

Bunu `docker-compose.override.yml` ile yapmanın *temiz bir yolu yoktur*. Bu dosyadaki `weblate` düğümüne `build: .` ekleyebilirsiniz. Ancak daha sonra özel kalıbınız sisteminizde `"weblate/weblate"` olarak etiketlenir ve bu da soruna yol açabilir.

Bu nedenle, `docker-compose.yml` dosyasını doğrudan resmi depo <https://github.com/WeblateOrg/docker-compose> üzerinden değiştirmeden kullanmak ve `docker-compose.override.yml` ile genişletmek yerine, resmi `docker-compose.yml` dosyasının bir kopyasını oluşturmak ve kopyanızı `image: weblate/weblate` yerine `build: .` ile değiştirmek için düzenlemek isteyebilirsiniz.

`docker-compose` kullanırken kaynaktan kalıp oluşturma hakkında ayrıntılı bilgi almak için [Dosya oluşturma referansı](#) bölümüne bakın.

5. Ayarları tanımlamak veya yeniden tanımlamak için özel ayarlar modülünüzü genişletin.

Hangi ayarların öncelikli olduğunu belirlemek için ayarları yukarıdaki içe aktarma ifadesinden önce ya da sonra tanımlayabilirsiniz. İçe aktarma ifadesinden önce tanımlanmış ayarlar, ortam değişkenleri ve veri biriminde tanımlanmış ayar değişiklikleri tarafından değiştirilebilir. İçe aktarma ifadesinden sonra tanımlanan ayar değiştirilmez.

Ayrıca daha ileri gidebilirsiniz. Örneğin, ayarların karşılığını belirlemek gibi `weblate.docker_settings` uygulamasının https://github.com/WeblateOrg/weblate/blob/main/weblate/settings_docker.py yaptığı bazı şeyleri yeniden oluşturabilirsiniz. Ortam değişkenleri olarak veya veri birimindeki Python dosyalarından ayarların değiştirilmesini sağlayabilirsiniz.

Logo ve diğer durağan dosyaları değiştirmek

3.8-5 sürümünde geldi.

Weblate ile birlikte gelen durağan dosyalar `/app/data/python/customize/static` içinden değiştirilebilir (bilgi almak için: [Docker kapsayıcısı birimleri](#)). Örneğin, `/app/data/python/customize/static/favicon.ico` dosyası favicon simgesi için kullanılır.

İpucu: Dosyalar kapsayıcı başlatıldığında ilgili konuma kopyalanır. Bu nedenle birimin içeriğini değiştirdikten sonra Weblate yeniden başlatılmalıdır.

Bu yaklaşım, Weblate kalıplarını değiştirmek için de kullanılabilir. Örneğin [Yasal](#) belgeler `/app/data/python/customize/templates/legal/documents` içine yerleştirilebilir.

Alternatif olarak, kendi modülünüzü de ekleyebilir (bilgi almak için: `../customize`) ve Docker kapsayıcısına ayrı birim olarak ekleyebilirsiniz. Örneğin:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ../weblate_customization/weblate_customization:/app/data/python/weblate_
      ↳ customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

PostgreSQL sunucusunu yapılandırmak

PostgreSQL kapsayıcısı varsayılan PostgreSQL yapılandırmasını kullanır ve işlemci çekirdeklerini ya da belleği etkili bir şekilde kullanmaz. Başarımı artırmak için yapılandırmanın özelleştirilmesi önerilir.

Yapılandırma, https://hub.docker.com/_/postgres adresindeki *Database Configuration* bölümünde açıklandığı gibi ayarlanabilir. Ortamınıza uygun bir yapılandırma oluşturmak için <https://pgtune.leopard.in.ua/> kullanabilirsiniz.

Kapsayıcı iç yapısı

Kapsayıcı, bireysel hizmetleri başlatmak için `program: supervisor` kullanır. [Yatay ölçeklendirme](#) kullanılması durumunda, bir kapsayıcıda yalnızca tek bir hizmet başlatılır.

Hizmetlerin durumunu denetlemek için şunu kullanın:

```
docker-compose exec --user weblate weblate supervisorctl status
```

Her Celery kuyruğu için ayrı hizmetler bulunur (bilgi almak için: `ref:celery`). Uygun işlemi durdurarak bazı görevlerin yürütülmesini durdurabilirsiniz:

```
docker-compose exec --user weblate weblate supervisorctl stop celery-translate
```

Debian ve Ubuntu üzerine kurmak

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve web sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
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
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

```
apt install -y \
  tesseract-ocr libtesseract-dev libleptonica-dev \
  libldap2-dev libldap-common libsasl2-dev \
  libxmlsec1-dev
```

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yönergeleri:

```
# 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

# Caching backend: Redis
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
apt install -y redis-server

# Database server: PostgreSQL
apt install -y postgresql postgresql-contrib

# SMTP server
apt install -y exim4
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen [İsteğe bağlı bağımlılıklar](#) seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (`. ~/weblate-env/bin/activate` komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu `~/weblate-env/bin/weblate` olarak belirtin.

1. `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` dosyasını `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py` dosyası olarak kopyalayın.
2. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veritabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: [Yapılandırmayı ayarlama](#).

3. Weblate için veritabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz):

```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adı ve web sitesini belirtmeniz gerekir.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Bileşen yapılandırması](#) for more details.

The important fields here are: [Bileşen adı](#), [Kaynak kod deposu](#), and [Dosya maskesi](#) for finding translatable files. Weblate supports a wide range of formats including [GNU gettext](#), [Android string resources](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Sstringsdict biçimi](#) or [Fluent biçimi](#), see [Desteklenen dosya biçimleri](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

SUSE ve openSUSE üzerine kurmak

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve web sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çevirler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
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
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libsasl2-devel
zypper install libxmlsec1-devel
```

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yönergeleri:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen *İsteğe bağlı bağımlılıklar* seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (`. ~/weblate-env/bin/activate` komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu `~/weblate-env/bin/weblate` olarak belirtin.

1. `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` dosyasını `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py` dosyası olarak kopyalayın.
2. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veritabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: *Yapılandırmayı ayarlama*.

3. Weblate için veritabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz):

```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adı ve web sitesini belirtmeniz gerekir.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Bileşen yapılandırması](#) for more details.

The important fields here are: [Bileşen adı](#), [Kaynak kod deposu](#), and [Dosya maskesi](#) for finding translatable files. Weblate supports a wide range of formats including [GNU gettext](#), [Android string resources](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Sstringsdict biçimi](#) or [Fluent biçimi](#), see [Desteklenen dosya biçimleri](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

RedHat, Fedora ve CentOS üzerine kurmak

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve web sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
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
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
dnf install libldap2-devel libsasl2-devel
dnf install libxmlsec1-devel
```

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yöntemleri:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen [İsteğe bağlı bağımlılıklar](#) seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (`. ~/weblate-env/bin/activate` komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu `~/weblate-env/bin/weblate` olarak belirtin.

1. `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` dosyasını `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py` dosyası olarak kopyalayın.
2. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veritabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: [Yapılandırmayı ayarlama](#).

3. Weblate için veritabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz):

```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adı ve web sitesini belirtmeniz gerekir.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Bileşen yapılandırması](#) for more details.

The important fields here are: [Bileşen adı](#), [Kaynak kod deposu](#), and [Dosya maskesi](#) for finding translatable files. Weblate supports a wide range of formats including [GNU gettext](#), [Android string resources](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Sstringsdict biçimi](#) or [Fluent biçimi](#), see [Desteklenen dosya biçimleri](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

macOS üzerine kurulum

Donanım gereksinimleri

Weblate, herhangi bir çağdaş donanımda sorunsuz çalışmalıdır, Tek bir sunucuda Weblate bileşenlerini (Weblate, veri tabanı ve web sunucusu) çalıştırmak için gerekli en az yapılandırma şu şekildedir:

- 3 GB RAM bellek
- 2 işlemci çekirdeği
- 1 GB depolama alanı

Bellek ne kadar çok olursa o kadar iyidir. Tüm düzeylerde (dosya sistemi, veri tabanı ve Weblate) ön bellek için kullanılır.

Eşzamanlı bulunacak kullanıcı sayısı, gerekli işlemci çekirdeği sayısını artırır. Yüzlerce bileşenden oluşan çeviriler için en az 4 GB RAM bellek bulunması önerilir.

Veri tabanı depolama alanı gereksinimi, barındırılacak 1 milyon sözcük için yaklaşık 300 MB olarak düşünebilir. Kopyalanmış depolar için gereken depolama alanı değişiklik gösterir. Ancak Weblate sığ kopyalama ile depo boyutlarını en az düzeyde tutmaya çalışır.

Not: Weblate kurulumunuz için gerçek gereksinimler, üzerinde bulunacak çevirilerin boyutuna bağlı olarak büyük ölçüde değişiklik gösterebilir.

Kurulum

Sistem gereksinimleri

Python modüllerini oluşturmak için gereken bağımlılıkları kurun (bilgi almak için: [Yazılım gereksinimleri](#)):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip install virtualenv
```

Make sure pip will be able to find the libffi version provided by homebrew — this will be needed during the installation build step.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig"
```

Kullanmak istediğiniz özelliklere bağlı olarak gereken isteğe bağlı bağımlılıkları kurun (bilgi almak için: [İsteğe bağlı bağımlılıklar](#)):

```
brew install tesseract
```

İsteğe bağlı olarak üretim sunucusunu çalıştırmak için gereken yazılımı kurun. Bilgi almak için: sunucu, [Weblate için veri tabanı kurulumu](#), [Celery ile arka plan görevlerini kullanmak](#). Kurulumunuzun boyutuna bağlı olarak, bu bileşenleri adanmış sunucular üzerinde çalıştırmak isteyebilirsiniz.

Yerel kurulum yönergeleri:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Database server: PostgreSQL
brew install postgresql
```

Python modülleri

İpucu: Sisteminizden ayrı bir ortama Weblate kurmak için virtualenv kullanıyoruz. Bu konuda bilginiz yoksa, virtualenv [User Guide](#) bölümüne bakabilirsiniz.

1. Weblate için virtualenv oluşturun:

```
virtualenv ~/weblate-env
```

2. Weblate için virtualenv etkinleştirin:

```
. ~/weblate-env/bin/activate
```

3. Tüm isteğe bağlı bağımlılıklar ile birlikte Weblate kurulumunu yapın:

```
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

İsteğe bağlı bağımlılıkları ayarlamak için lütfen [İsteğe bağlı bağımlılıklar](#) seçeneğini denetleyin.

Not: Bazı Linux dağıtımlarında Weblate çalıştırıldığında libffi hatasıyla karşılaşılabilir:

```
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)
```

Bu durum, PyPI ile dağıtılan binary paketlerinin dağıtımla uyumsuz olmasından kaynaklanır. Bu sorunu gidermek için, paketi sisteminizde yeniden oluşturmanız gerekir:

```
pip install --force-reinstall --no-binary :all: cffi
```

Weblate platformunu yapılandırmak

Not: Aşağıdakiler, Weblate için virtualenv kullanımının etkinleştirilmiş olduğunu varsayar (`. ~/weblate-env/bin/activate` komutu yürütülerek). Virtualenv etkin değilse, **weblate** komutunun tam yolunu `~/weblate-env/bin/weblate` olarak belirtin.

1. `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` dosyasını `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py` dosyası olarak kopyalayın.
2. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veritabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: [Yapılandırmayı ayarlama](#).
3. Weblate için veritabanını ve yapısını oluşturun (örnek ayarlar PostgreSQL kullanır, üretime hazır bir kurulum için [Weblate için veri tabanı kurulumu](#) bölümüne bakabilirsiniz):

```
weblate migrate
```

4. Yönetici için bir hesap oluşturun ve parolasını panoya kopyalayıp daha sonra kullanmak üzere kaydedin:

```
weblate createadmin
```

5. Web sunucunuz için durağan dosyaları derleyin (bilgi almak için: [Sunucuyu çalıştırmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate collectstatic
```

6. JavaScript ve CSS dosyalarını sıkıştırın (isteğe bağlı, bilgi almak için: [İstemci varlıklarının sıkıştırılması](#)):

```
weblate compress
```

7. Celery işlemlerini başlatın. Bu adım, geliştirme ortamları için gerekli değildir. Ancak üretim ortamlarında yapılması önemle önerilir. Bilgi almak için [Celery ile arka plan görevlerini kullanmak](#):

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Geliştirme sunucusunu başlatın ([Sunucuyu çalıştırmak](#) bölümünden üretim kurulumunun ayrıntılarına bakabilirsiniz):

```
weblate runserver
```

Kurulum sonrası

Tebrikler, Weblate sunucunuz artık çalışıyor ve kullanmaya başlayabilirsiniz.

- Weblate arayüzüne <http://localhost:8000/> adresinden erişebilirsiniz.
- Kurulum sırasında belirtilen yönetici kimlik doğrulama bilgileriyle oturum açın ya da yeni kullanıcı hesapları açın.
- Weblate virtualenv etkinleştirildiğinde **weblate** komutunu kullanarak Weblate komutlarını yürütebilirsiniz, Bilgi almak için: [Yönetim komutları](#).
- Deneme sunucusunu Ctrl+C tuşlarıyla durdurabilirsiniz.
- Kurulumunuzla ilgili olası sorunları [/manage/performance/](#) adresinden (bilgi almak için: [Yönetim arayüzü](#)) ya da **weblate check --deploy** komutundan görebilirsiniz. Bilgi almak için: [Üretim kurulumu](#).

Çeviri eklemek

1. Yönetim arayüzünü açın (<http://localhost:8000/create/project/>) ve çevirmek istediğiniz bir proje oluşturun. Bilgi almak için [Proje yapılandırması](#).

Burada yalnızca proje adı ve web sitesini belirtmeniz gerekir.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See [Bileşen yapılandırması](#) for more details.

The important fields here are: [Bileşen adı](#), [Kaynak kod deposu](#), and [Dosya maskesi](#) for finding translatable files. Weblate supports a wide range of formats including [GNU gettext](#), [Android string resources](#), [Apple iOS dizgeleri](#), [Java özellikleri](#), [Stringsdict biçimi](#) or [Fluent biçimi](#), see [Desteklenen dosya biçimleri](#) for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Kaynaklardan kurulum

1. Lütfen Weblate kurulumu için öncelikle sisteminize uygun kurulum yönergelerini izleyin:

- *Debian ve Ubuntu üzerine kurmak*
- *SUSE ve openSUSE üzerine kurmak*
- *RedHat, Fedora ve CentOS üzerine kurmak*

2. Git kullanarak güncel Weblate kaynaklarını edinin (ya da bir tarball indirip paketi ayıklayın):

```
git clone https://github.com/WeblateOrg/weblate.git weblate-src
```

Alternatif olarak, yayınlanmış arşiv dosyalarını da kullanabilirsiniz. Bunları web sitemizden <https://weblate.org/> indirebilirsiniz. Bu indirmeler şifrelenerek imzalanmıştır. Bilgi almak için: ref: 'verify'.

3. Geçerli Weblate kodunu virtualenv üzerine kurun:

```
. ~/weblate-env/bin/activate  
pip install -e weblate-src
```

4. `weblate/settings_example.py` dosyasını `weblate/settings.py` olarak kopyalayın.
5. Yeni `settings.py` dosyasındaki değerleri istediğiniz gibi ayarlayın. En azından veritabanı kimlik doğrulama bilgilerini ve Django gizli anahtarını belirtmeniz gerekir. Ancak üretim kurulumu için daha fazla değişiklik yapmak isteyeceksiniz. Bilgi almak için: [Yapılandırma ayarlama](#).
6. Weblate tarafından kullanılacak veri tabanını oluşturun. Bilgi almak için: [Weblate için veri tabanı kurulumu](#).
7. Django tablolarını, durağan dosyaları ve başlangıç verilerini oluşturun (bilgi almak için: [Veri tabanını doldurmak](#) ve [Durağan dosyalar sunmak](#)):

```
weblate migrate  
weblate collectstatic  
weblate compress
```

Not: Bu adım, her depo güncellemesi yaptığınızda yinelenmelidir.

OpenShift üzerine kurulum

OpenShift Weblate kalıbıyla kişisel Weblate bulut sunucunuzu saniyeler içinde çalışır duruma getirebilirsiniz. Tüm Weblate bağımlılıkları önceden katılmıştır. PostgreSQL varsayılan veritabanı olarak ayarlanır ve kalıcı birim kullanılır.

Kalıbı <https://github.com/WeblateOrg/openshift/> adresinde bulabilirsiniz.

Kurulum

Aşağıdaki örneklerde, `oc` istemci aracının yüklü olduğu çalışan bir OpenShift v3.x ortamınız olduğu varsayılmıştır. Lütfen yönergeler için OpenShift belgelerine bakın.

`template.yml` dosyası OpenShift üzerinde tüm bileşenleri çalıştırmak için uygundur. Ayrıca `template-external-postgresql.yml` dosyası bir PostgreSQL sunucusunu başlatmaz ve bir dış PostgreSQL sunucusunu yapılandırmanızı sağlar.

Web konsolu

Ham içeriği `template.yml` dosyasından kopyalayın ve projenize aktarın. Ardından uygulamanızı oluşturmak için OpenShift web konsolundaki `Create` düğmesini kullanın. Web konsolu, sizden kalıp tarafından kullanılan tüm parametrelerin değerlerini ister.

CLI

Weblate kalıbını geçerli projenizin kalıp kitaplığına yüklemek için şu komutta `template.yml` dosyasını belirtin:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
→template.yml \
  -n <PROJECT>
```

Kalıp artık web konsolu ya da komut satırı (CLI) kullanılarak seçilebilir.

Parametreler

Değiştirebileceğiniz parametreler kalıbın parametreler bölümünde listelenir. Şu komutu kullanarak ve kullanılacak dosyayı belirterek bunları komut satırı (CLI) ile listeleyebilirsiniz:

```
$ 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
```

Hazırlamak

Komut satırını (CLI) kalıpları işlemek ve hazırlanan yapılandırmadan nesneleri hemen oluşturmak için de kullanabilirsiniz.

```
$ 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
```

Weblate kopyası, başarılı bir aktarım ve dağıtımdan sonra belirtilmiş `WEBLATE_SITE_DOMAIN` parametresinde kullanılabilir olmalıdır.

Kapsayıcı kurulumundan sonra, `WEBLATE_ADMIN_PASSWORD` seçeneğinde belirtilen parolayla ya da ayarlanmamışsa ilk başlatılmada oluşturulan rastgele bir parolayla `admin` kullanıcısı olarak oturum açabilirsiniz.

‘Admin’ parolasını sıfırlamak için, ilgili `Secret` bölümünde `WEBLATE_ADMIN_PASSWORD` seçeneğine yeni parola ayarladıktan sonra kapsayıcıyı yeniden başlatın.

Elemek

```
$ 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
```

Yapılandırma

Kalıbı işleyerek, Weblate kalıbını özelleştirmek için kullanılacak ilgili bir ConfigMap oluşturulur. ConfigMap doğrudan ortam değişkenleri olarak bağlanır ve her değiştirildiğinde yeni bir dağıtımı tetikler. Daha fazla yapılandırma seçeneği sunan ortam değişkenlerinin tam listesi için *Docker ortam değişkenleri* bölümüne bakabilirsiniz.

Kubernetes üzerine kurulum

Not: Bu rehberde, kurulumu daha ayrıntılı olarak ele alabilmek için Kubernetes konusunda deneyimli gönüllüler arıyoruz.

Kubernetes Helm çizelgesi ile kişisel Weblate bulut sunucunuzu saniyeler içinde çalışır duruma getirebilirsiniz. Tüm Weblate bağımlılıkları önceden katılmıştır. Varsayılan veritabanı olarak PostgreSQL ayarlanır ve kalıcı birim kullanılır.

Çizelgeyi <<https://github.com/WeblateOrg/helm/>> adresinde bulabilir ve <<https://artifacthub.io/packages/helm/weblate/weblate>> adresinde görebilirsiniz.

Kurulum

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Yapılandırma

Daha fazla yapılandırma seçeneği sunan ortam değişkenlerinin tam listesi için *Docker ortam değişkenleri* bölümüne bakabilirsiniz.

Kurulumunuza ve deneyiminize bağlı olarak, size en uygun kurulum yöntemi seçin:

- Üretim kurulumları için önerilen *Docker ile kurmak*.
- Üretim kurulumları için önerilen virtualenv kurulumu:
 - *Debian ve Ubuntu üzerine kurmak*
 - *SUSE ve openSUSE üzerine kurmak*
 - *RedHat, Fedora ve CentOS üzerine kurmak*
 - *macOS üzerine kurulum*

- *Kaynaklardan kurulum*, geliştirme çalışmaları için önerilir.
- *OpenShift üzerine kurulum*
- *Kubernetes üzerine kurulum*

2.1.2 Yazılım gereksinimleri

İşletim sistemi

Weblate platformunun, Linux, FreeBSD ve macOS üzerinde çalıştığı biliniyor. Diğer Unix benzeri sistemlerde de büyük olasılıkla çalışacaktır.

Weblate, Windows üzerinde desteklenmez. Ancak yine de çalışabilir ve yazılım yamaları mutlulukla kabul edilir.

Diğer hizmetler

Weblate, çalışabilmek için başka hizmetleri kullanır. En azından şu hizmetlerin çalışması gerekir:

- PostgreSQL veri tabanı sunucusu. Bilgi almak için: *Weblate için veri tabanı kurulumu*.
- Ön bellek ve görev kuyruğu için Redis sunucusu. Bilgi almak için: *Celery ile arka plan görevlerini kullanmak*.
- Giden e-postalar için kullanılacak SMTP sunucusu. Bilgi almak için: *Giden e-postayı yapılandırmak*.

Python bağımlılıkları

Weblate, Python ile yazılmıştır ve Python 3.6 ya da üzerindeki sürümleri destekler. Bağımlılıkları pip kullanarak ya da dağıtım paketlerinizden kurabilirsiniz. Tam listeyi `requirements.txt` adresinde bulabilirsiniz.

En önemli bağımlılıklar:

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 çatısı

<https://www.django-rest-framework.org/>

İsteğe bağlı bağımlılıklar

Bazı Weblate özellikleri için şu modüller gereklidir. Hepsini `requirements-optional.txt` içinde bulabilirsiniz.

Mercurial (*Mercurial* depo desteği için isteğe bağlı)

<https://www.mercurial-scm.org/>

phply (*PHP dizgeleri* için isteğe bağlı)

<https://github.com/viraptor/phply>

tesseract (*Visual context for strings* ile karakter tanıma için isteğe bağlı)

<https://github.com/sirfz/tesseract>

python-akismet (*Spam protection için isteğe bağlı*)

<https://github.com/Nekmo/python-akismet>

ruamel.yaml (*YAML dosyaları için isteğe bağlı*)

<https://pypi.org/project/ruamel.yaml/>

Zeep (*Microsoft Terminology için isteğe bağlı*)

<https://docs.python-zeep.org/>

aeidon (*Alt yazı dosyaları için isteğe bağlı*)

<https://pypi.org/project/aeidon/>

fluent.syntax (*Fluent biçimi için isteğe bağlı*)

<https://projectfluent.org/>

İpucu: Pip kullanarak kurulum yapılırken istediğiniz özellikleri doğrudan belirtebilirsiniz:

```
pip install "Weblate[PHP,Fluent]"
```

Ya da tüm isteğe bağlı özelliklerle Weblate kurulumu yapabilirsiniz:

```
pip install "Weblate[all]"
```

Ya da hiç bir isteğe bağlı özellik olmadan Weblate kurulumu yapabilirsiniz:

```
pip install Weblate
```

Veri tabanı arka plan bağımlılıkları

Weblate PostgreSQL, MySQL ve MariaDB veri tabanlarını destekler. Bilgi almak için *Weblate için veri tabanı kurulumu* ve yönetim bölümü belgelerine bakabilirsiniz.

Diğer sistem gereksinimleri

Sisteme şu bağımlılıkların kurulması gerekir:

Git

<https://git-scm.com/>

Pango, Cairo ve ilişkili üst bilgi dosyaları ile GObject iç gözlem verileri

<https://cairographics.org/>, <https://pango.gnome.org/>, bilgi almak için: *Pango ve Cairo*

git-review (Gerrit desteği için isteğe bağlı)

<https://pypi.org/project/git-review/>

git-svn (Subversion desteği için isteğe bağlı)

<https://git-scm.com/docs/git-svn>

tesseract ve verileri (ekran görüntülerinde karakter tanıma için isteğe bağlı)

<https://github.com/tesseract-ocr/tesseract>

licensee (bileşen oluştururken lisansın algılanması için isteğe bağlı)

<https://github.com/licensee/licensee>

Yapım zamanı bağımlılıkları

Bazı *Python bağımlılıkları* bağımlılıklarını kurmanız gerekebilir. Bu durum, bunları nasıl kurduğunuza bağlıdır. O yüzden her paketin kendi belgesine başvurun. `Pip` kullanarak yapılan kurulum sırasında ya da dağıtım paketlerini kullanırken önceden oluşturulmuş `Wheels` kullanıyorsanız, bunlara gerek duymazsınız.

Pango ve Cairo

3.7 sürümünde değişti.

Weblate, bitmap bileşenlerini (bilgi almak için: *promotion*) ve görüntüleme denetimlerini (bilgi almak için: *Yazı tiplerini yönetmek*) oluşturmak için Pango ve Cairo kullanır. Python bağlantılarını düzgün olarak kurmak için önce sistem kitaplıklarını kurmanız gerekir. Hem Cairo hem de Pango gereklidir, bunun için de GLib gereklidir. Tüm bunlar geliştirme dosyaları ve GObject içi gözlem verileriyle birlikte kurulmalıdır.

2.1.3 Sürüm imzalarını doğrulamak

Weblate sürümü, yayın geliştiricisi tarafından şifrelenmiş olarak imzalanır. Şu anda bu işi Michal Čihař yapıyor ve onun PGP anahtarının parmak izi:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

ve <https://keybase.io/nijel> adresinden diğer kimlik bilgilerini alabilirsiniz.

İmzanın indirdiğiniz arşivle eşleştiğini doğrulamanız gerekir. Bu şekilde, yayınlanan kodu değiştirilmemiş olduğundan emin olabilirsiniz. Güncel sürümü indirdiğinizden emin olmak için imzanın tarihini de doğrulamalısınız.

Her arşiv dosyası için, PGP imzasının bulunduğu `.asc` dosyaları bulunur. Bu iki dosyayı aynı klasöre kaydettikten sonra imzayı doğrulayabilirsiniz:

```
$ 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:
gpg: using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

Görebileceğiniz gibi GPG, herkese açık anahtarı bilmediğini bildiriyor. Bu aşamada şu adımlardan birini uygulamanız gerekir:

- Anahtarı indirmek için `wkd` kullanın:

```
$ 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]
```

- Michal sunucusundan anahtarlığı indirip şununla içe aktarın:

```
$ gpg --import wxth3chu9jfxdxywj1skpmhsj311mzm
```

- Anahtarı bir anahtar sunucusundan indirip içe aktarın:

```
$ 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
```

Bu uygulama, durumu biraz iyileştirir. Bu noktada, belirtilen anahtardaki imzanın doğruluğundan emin olabilirsiniz. Ancak yine de anahtarda kullanılan ada güvenemezsiniz:

```
$ 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
```

Buradaki sorun, herhangi birinin aynı adlı bir anahtar yayınlayabilmesidir. Anahtarın gerçekten ilgili kişinin olduğundan emin olmanız gerekir. Bu konu, GNU gizlilik rehberinin [Herkes açık anahtarlığınızda diğer anahtarları doğrulama](#) bölümünde ele alınmıştır. En güvenilir yöntem, geliştiriciyle birebir tanışmak ve anahtar parmak izlerini takas etmektir. Ancak güven ağına da güvenebilirsiniz. Bu şekilde, geliştiriciyle birebir tanışmış başka kişilerin imzalarıyla anahtara geçici olarak güvenebilirsiniz.

Anahtara güvenildikten sonra uyarı görünmez olur:

```
$ 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]
```

İmza geçersiz olursa (arşiv değiştirilmişse), anahtara güvenilirlik güvenilmediğine bakılmaksızın açık bir hata iletisi görürsünüz:

```
$ 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 Dosya sistemi izinleri

Weblate işleminin, verileri tuttuğu `DATA_DIR` klasöründe okuma ve yazma yapabilmesi gerekir. Bu klasördeki tüm dosyaların sahibi, tüm Weblate işlemlerini çalıştıran kullanıcı olmalı ve bu klasöre yazabilmelidir (genellikle WSGI ve Celery, bilgi almak için sunucu ve [Celery ile arka plan görevlerini kullanmak](#)).

Varsayılan yapılandırmada bunlar Weblate kaynaklarıyla aynı ağaca yerleştirilir. Ancak bunları `/var/lib/weblate` gibi daha iyi bir konuma taşımayı yeğleyebilirsiniz.

Weblate bu klasörleri kendiliğinden oluşturmaya çalışır. Ancak bunu yapmak için yeterli izinleri yoksa bunu yapamaz.

[Yönetim komutları](#) komutunu çalıştırırken de dikkatli olmalısınız. Komut aynı Weblate kullanıcısı ile çalıştırılmalıdır yoksa bazı dosyaların izinler yanlış olabilir.

Docker kapsayıcısında, /app/data birimindeki tüm dosyaların sahibi kapsayıcı içindeki weblate kullanıcısı olmalıdır (UID 1000).

Ayrıca bakınız:

Durağan dosyalar sunmak

2.1.5 Weblate için veri tabanı kurulumu

Weblate için PostgreSQL veri tabanı sunucusunun kullanılması önerilir.

Ayrıca bakınız:

Güç bir veri tabanı sunucusu kullanın, Databases, Diğer veri tabanlarından PostgreSQL üzerine aktarım

PostgreSQL

Django temelli siteler için genellikle en iyi seçim PostgreSQL kullanmaktır. Django tarafından referans olarak kullanılan veri tabanıdır.

Not: Weblate, bazı durumlarda ayrı olarak kurulması gereken trigram eklentisini kullanır. `Postgresql-contrib` ya da benzer şekilde adlandırılmış bir paket arayın.

Ayrıca bakınız:

PostgreSQL notes

PostgreSQL üzerinde bir veri tabanı oluşturmak

Weblate için ayrı bir kullanıcı hesabı ile ayrı bir veri tabanı kullanmak genellikle iyi bir fikirdir:

```
# 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
```

İpucu: Weblate kullanıcılarını PostgreSQL üzerinde süper kullanıcı yapmak istemiyorsanız, bu seçeneği atlayabilirsiniz. Bu durumda, PostgreSQL süper kullanıcısı ile Weblate şemasındaki bazı aktarım adımlarını el ile yapmanız gerekir:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
CREATE EXTENSION IF NOT EXISTS btree_gin WITH SCHEMA weblate;
```

Weblate yapılandırmasını PostgreSQL kullanacak biçimde ayarlamak

PostgreSQL için `settings.py` kod parçası:

```
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": "",
    }
}
```

Veri tabanı aktarımı, Weblate tarafından kullanılan **ALTER ROLE** veri tabanı rolüyle gerçekleştirir. Çoğu durumda rolün adı kullanıcı adıdır. Daha karmaşık kurulumlarda rol adı kullanıcı adından farklıdır ve veri tabanı aktarımı sırasında rolün var olmadığı ile ilgili bir hata iletisi görürsünüz (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Bu sorunun PostgreSQL için Azure veri tabanı ile ortaya çıktığı biliniyor. Ancak yalnızca bu durumla sınırlı değildir. Lütfen veri tabanı aktarımı sırasında Weblate tarafından kullanılacak rolün adını **ALTER_ROLE** seçeneğinden ayarlayın.

MySQL ve MariaDB

İpucu: Bazı Weblate özellikleri *PostgreSQL* ile daha iyi çalışır. PostgreSQL veri tabanı, tam metin özelliklerinin kullanılmasını sağlar ve arama ile çeviri belleği işlemlerinde daha üstündür.

Weblate, MySQL ya da MariaDB ile de kullanılabilir. Lütfen bunları Django ile kullanmakla ilgili uyarılar için [MySQL notes](#) ve [MariaDB notes](#) bölümlerine bakın. Bazı sınırlamalar nedeniyle, yeni kurulumlar için *PostgreSQL* kullanılması önerilir.

Weblate için en az MySQL 5.7.8 ya da MariaDB 10.2.7 sürümü kullanılmalıdır.

Weblate için önerilen yapılandırma:

- Daha yüksek Unicode düzlemlerinin (emojiler gibi) görüntülenmesini sağlamak için `utf8mb4` karakter kümesini kullanın.
- Metin alanlarında daha uzun dizinlerin kullanılabilmesini sağlamak için sunucuyu `innodb_large_prefix` seçeneği ile yapılandırın.
- Yalıtım düzeyini `READ COMMITTED` olarak ayarlayın.
- SQL kipi `STRICT_TRANS_TABLES` olarak ayarlanmalıdır.

MySQL 8.x, MariaDB 10.5.x ya da üzeri için varsayılan yapılandırma yeterlidir. Bu nedenle sunucunun ayarlanması gerekmez ve gereken her şey istemci tarafında yapılandırılabilir.

Aşağıda 8 GB RAM belleği olan bir sunucu için `/etc/my.cnf.d/server.cnf` dosyasının örneğini bulabilirsiniz. Bu ayarlar çoğu kurulum için yeterli olmalıdır. MySQL ve MariaDB için, sisteme aynı anda çok sayıda kullanıcının erişmesinin gerekmediği durumlarda sunucu başarımını artıracak seçenekler vardır. Ayrıntılı bilgiyi üretici belgelerinde bulabilirsiniz.

Weblate kurulumunuza başlamadan önce `innodb_file_per_table` seçeneğinin doğru ayarlanması ve MySQL ya da MariaDB veri tabanı sunucusunun yeniden başlatılması kurulum yaparken karşılaşılabilecek sorunları azaltmak için çok önemlidir.

```
[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
```

İpucu: #1071 - Specified key was too long; max key length is 767 bytes hatasını görürseniz, yapılandırmanızı yukarıdaki `innodb` ayarlarına uygun olarak güncelleyin ve kurulumu yeniden başlatın.

İpucu: #2006 - MySQL server has gone away hatasını görürseniz, `CONN_MAX_AGE` seçeneğini yapılandırmak yardımcı olabilir.

Weblate için MySQL/MariaDB yapılandırması

MySQL ve MariaDB için `settings.py` kod parçası:

```
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": {},
    }
}
```

Kurulumu başlamadan önce MySQL ya da MariaDB veri tabanı sunucusu üzerinde `weblate` kullanıcı hesabını oluşturmanız gerekir. Bunun için şu komutları kullanın:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Diğer yapılandırmalar

Giden e-postayı yapılandırmak

Weblate, hesap etkinleştirme işlemleri ve kullanıcılar tarafından yapılandırılmış çeşitli bildirimler gibi çeşitli durumlar için e-posta gönderir. Bunun için bir SMTP sunucu erişiminin yapılandırılması gerekir.

E-posta sunucusu şu ayarlar kullanılarak yapılandırılır: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_TLS`, `EMAIL_USE_SSL`, `EMAIL_HOST_USER` ve `EMAIL_PORT`. Seçeneklerin adları oldukça açıktır. Ayrıntılı bilgi almak için Django belgelerine bakabilirsiniz.

İpucu: Kimlik doğrulanmasının desteklenmediği ile ilgili bir hata görürseniz (SMTP AUTH extension not supported by server gibi), bu sorun büyük olasılıkla güvenli olmayan bağlantı kullanımından kaynaklanıyordur ve sunucu kimliğinin bu şekilde doğrulanmasını reddeder. Böyle bir durumda `EMAIL_USE_TLS` seçeneğini etkinleştirmeyi deneyin.

Ayrıca bakınız:

Not receiving e-mails from Weblate, Docker kapsayıcısında giden e-posta ayarlarını yapılandırmak

Ters vekil sunucu arkasında çalıştırmak

Hız sınırlama, *Spam protection* ve *Denetim günlüğü* Weblate özellikleri istemci IP adresinin bilinmesine dayanır.

Varsayılan yapılandırmada Weblate, WSGI işleyicisi tarafından ayarlanan `REMOTE_ADDR` IP adresini alır.

Ters vekil sunucu kullanıyorsanız, bu alanda büyük olasılıkla ters vekil sunucu adresi bulunacaktır. Weblate yapılandırmasını ek HTTP üst bilgilerine güvenecek ve IP adresini bunlardan alacak şekilde ayarlamanız gerekir. Bu yapılandırma, ters vekil sunucu kullanmayan kurulumlar için IP adresi sahteciliğine olanak sağlayacağından varsayılan olarak etkinleştirilemez. `IP_BEHIND_REVERSE_PROXY` seçeneğini etkinleştirmek genel kurulumlar için yeterli olabilir. Ancak `IP_PROXY_HEADER` ve `IP_PROXY_OFFSET` seçeneklerini de ayarlamanız gerekebilir.

Dikkat edilmesi gereken başka bir şey de `Host` üst bilgisidir ve `SITE_DOMAIN` olarak yapılandırılmış değerle eşleşmelidir. Ters vekil sunucunuzda ek yapılandırma gerekebilir (örneğin, Apache için `ProxyPreserveHost On` ya da nginx ile `proxy_set_header Host $host;` kullanın).

Ayrıca bakınız:

Spam protection, *Hız sınırlama*, *Denetim günlüğü*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

HTTP vekil sunucu

Weblate, sürüm denetimi sistemi komutlarını yürütür ve bunlar ortamın vekil sunucu yapılandırmasını alır. Önerilen yaklaşım, `settings.py` dosyasında vekil sunucu ayarlarını belirtmektir:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Ayrıca bakınız:

Vekil sunucu ortam değişkenleri

2.1.7 Yapılandırmayı ayarlama

Ayrıca bakınız:

Örnek yapılandırma

`weblate/settings_example.py` dosyasını `weblate/settings.py` dosyasına kopyalayın ve kurulumunuza uygun olarak ayarlayın. Büyük olasılıkla şu seçenekleri ayarlamak isteyeceksiniz: `ADMINS`

Bir şeyler ters gittiğinde bildirim alacak site yöneticilerinin listesi. Başarısız olan birleştirme bildirimleri ya da Django hataları gibi.

Ayrıca bakınız:

`ADMINS`, *Yöneticileri düzgün şekilde yapılandırın*

`ALLOWED_HOSTS`

Bu seçeneği, sitenizin sunması gereken barındırma hizmetlerini listeleyecek biçimde ayarlamalısınız. Örneğin:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternatif olarak genel arama karakteri ekleyebilirsiniz:

```
ALLOWED_HOSTS = ["*"]
```

Ayrıca bakınız:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Allowed hosts kurulumu*

`SESSION_ENGINE`

Oturumlarınızın nasıl kaydedileceğini yapılandırın. Varsayılan veri tabanı arka uç altyapısını korumanız durumunda, eski oturum verilerini veri tabanından kaldırmak için `weblate clearsessions` görevini zamanlamanız gerekir.

Ön bellek olarak Redis kullanıyorsanız (bilgi almak için: *Ön bellek özelliğini açın*) onu oturumlar için de kullanmanız önerilir:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Ayrıca bakınız:

Configuring the session engine, `SESSION_ENGINE`

`DATABASES`

Veri tabanı sunucusu ile bağlantı. Ayrıntı bilgi almak için Django belgelerine bakabilirsiniz.

Ayrıca bakınız:

Weblate için veri tabanı kurulumu, `DATABASES`, *Databases*

`DEBUG`

Herhangi bir üretim sunucusunda bu seçeneği devre dışı bırakın. Hata ayıklama kipi etkinleştirildiğinde, Django kullanıcılara hata bildirimlerini görüntüler. Devre dışı bırakıldığında, hatalar e-posta olarak `ADMINS` seçeneğindeki adreslere gönderilir (yukarıya bakın).

Hata ayıklama kipi de Weblate işleyişini yavaşlatır. Çünkü Django bu durumda içeride çok daha fazla bilgi depolar.

Ayrıca bakınız:

`DEBUG`, *Hata ayıklama kipini kapatın*

`DEFAULT_FROM_EMAIL`

Hesap açma e-postaları gibi giden e-postalar için e-posta gönderici adresi.

Ayrıca bakınız:

`DEFAULT_FROM_EMAIL`

`SECRET_KEY`

Django tarafından çerezlerdeki bazı bilgileri imzalamak için kullanılan anahtar. Bilgi almak için: [Django gizli anahtarı](#).

Ayrıca bakınız:

`SECRET_KEY`

`SERVER_EMAIL`

Başarısız olan birleştirme bildirimleri gibi, yöneticiye gönderilecek e-postalarda gönderici adresi olarak kullanılacak e-posta adresleri.

Ayrıca bakınız:

`SERVER_EMAIL`

2.1.8 Veri tabanını doldurmak

Yapılandırmanız hazır olduktan sonra, veri tabanı yapısını oluşturmak için `weblate migrate` komutunu çalıştırabilirsiniz. Bundan sonra yönetici arayüzünü kullanarak çeviri projeleri oluşturabilirsiniz.

Bir kurulumu etkileşimsiz olarak yapmak isterseniz, `weblate migrate --noinput` komutunu kullanabilir ve ardından `createadmin` komutu ile bir yönetici kullanıcı oluşturabilirsiniz.

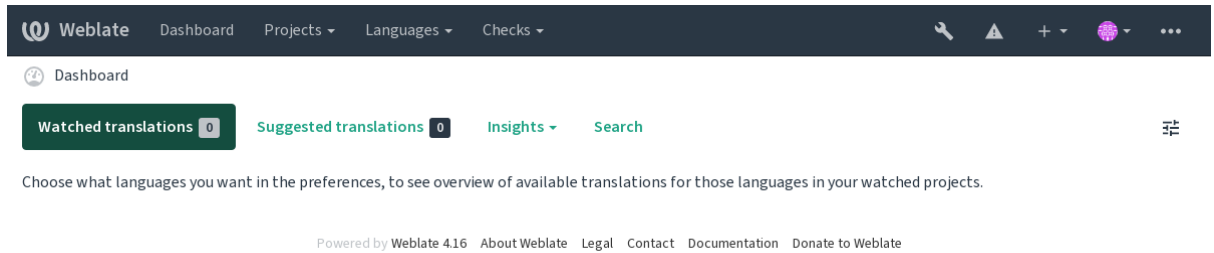
İşiniz bittiğinde, yönetici arayüzündeki `:guilabel:Başarım` raporu bölümüne bakmalısınız. Burada size site yapılandırmasını iyileştirmeniz için ipuçları sunulur.

Ayrıca bakınız:

[Yapılandırma, Yetkiler ve hazır rollerin listesi](#)

2.1.9 Üretim kurulumu

Bir üretim kurulumu için aşağıdaki bölümlerde açıklanan ayarlamaları yapmanız gerekir. Süper kullanıcı olarak oturum açıldığında en önemli ayarlar üst çubukta ünlem simgesi ile bir uyarı olarak görüntülenir:



Django tarafından tetiklenen denetimleri de incelemeniz önerilir (ancak hepsini düzeltmeniz gerekmez):

```
weblate check --deploy
```

Aynı denetim listesini [Yönetim arayüzü](#) bölümünden de gözden geçirebilirsiniz.

Ayrıca bakınız:

[Deployment checklist](#)

Hata ayıklama kipini kapatın

Şununla Django hata ayıklama kipini kaptın (*DEBUG*):

```
DEBUG = False
```

Hata ayıklama kipi açıkken, Django yürütülen tüm sorguları depolar ve kullanıcılara, üretim kipinde gerek duyulmayan hata izlerini görüntüler.

Ayrıca bakınız:

Yapılandırmayı ayarlama

Yöneticileri düzgün şekilde yapılandırın

Sunucuda bir sorun olması durumunda e-postaları kimlerin alacağını belirlemek için doğru yönetici adreslerini *ADMINS* seçeneği ile ayarlayın. Örneğin:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Ayrıca bakınız:

Yapılandırmayı ayarlama

Doğru site etki alanını ayarlayın

Yönetici arabiriminde site adını ve etki alanını ayarlayın. Yoksa RSS ya da kayıt e-postalarındaki bağlantılar çalışmaz. Bu ayar, site etki alanı adının yazılması gereken *SITE_DOMAIN* seçeneği ile yapılandırılır.

4.2 sürümünde değişti: 4.2 sürümünden önce bunun yerine Django site çatısı kullanılıyordu. Bilgi almak için [The “sites” framework](#).

Ayrıca bakınız:

Allowed hosts kurulumu, *HTTPS* ayarını düzgün biçimde yapın *SITE_DOMAIN*, *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

HTTPS ayarını düzgün biçimde yapın

Weblate için şifrelenmiş HTTPS iletişim kuralını kullanmanız önemle önerilir. Sertifikanızı hazırladıktan sonra *ENABLE_HTTPS* seçeneğini ayarlamanız gerekir:

```
ENABLE_HTTPS = True
```

İpucu: Ayrıca HSTS özelliğini de etkinleştirmek isteyebilirsiniz. Bilgi almak için: [SSL/HTTPS](#).

Ayrıca bakınız:

ENABLE_HTTPS, *Allowed hosts* kurulumu, *Doğru site etki alanını ayarlayın*

SECURE_HSTS_SECONDS seçeneğini düzgün biçimde ayarlayın

Siteniz SSL üzerinden sunuluyorsa, HTTP sıkı aktarım güvenliği (HTTP Strict Transport Security) özelliğini etkinleştirmek için `settings.py` dosyasında `SECURE_HSTS_SECONDS` değerini ayarlamayı değerlendirmeniz gerekir. Bu seçenek varsayılan olarak, aşağıda gösterildiği gibi 0 olarak ayarlanmıştır.

```
SECURE_HSTS_SECONDS = 0
```

Sıfır olmayan bir tamsayı değerine ayarlanırsa, `django.middleware.security.SecurityMiddleware` üst bilgisi bulunmayan tüm yanıtlarda **HTTP Strict Transport Security** üst bilgisini ayarlar.

Uyarı: Bu ayar yanlış yapılrısa, sitenizi geri dönüşü olmayan bir şekilde (bir süreliğine) bozabilir. Önce **HTTP Strict Transport Security** belgelerini okuyun.

Güç bir veri tabanı sunucusu kullanın

- Lütfen üretim ortamı için PostgreSQL kullanın. Bilgi almak için: *Weblate için veri tabanı kurulumu*.
- Veri tabanı sunucusunu çalıştırmak için yakın konumları kullanın. Yoksa ağ başarımı ya da güvenilirliği sorunları Weblate deneyiminizi mahvedebilir.
- Veritabanı sunucusunun başarımını denetleyin veya yapılandırmasını değiştirin. Örneğin **PGTune** kullanabilirsiniz.

Ayrıca bakınız:

Weblate için veri tabanı kurulumu, *Diğer veri tabanlarından PostgreSQL üzerine aktarım*, *Yapılandırmayı ayarlama*, *Databases*

Ön bellek özelliğini açın

Olabiliyorsa, CACHES yapılandırma değişkenini ayarlayarak Django Redis kullanın. Örneğin:

```
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',
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
}
```

İpucu: Ön bellek için Redis ayarlarını değiştirirseniz, bunları Celery için de ayarlamanız gerekebilir. Bilgi almak için: :ref:`Celery`.

Ayrıca bakınız:

Avatar ön belleği, Django's cache framework

Avatar ön belleği

Django ön belleğinin yanında, Weblate avatarları da ön belleğe alır. Bu amaçla ayrı, dosya temelli bir ön bellek kullanılması önerilir:

```
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,
        },
    },
}
```

Ayrıca bakınız:

[ENABLE_AVATARS](#), [AVATAR_URL_PREFIX](#), *Avatarlar*, *Ön bellek özelliğini açın*, Django's cache framework

E-posta gönderimini yapılandırın

Weblate tarafından bazı e-postaların gönderilmesi gerekir ve bu e-postaların doğru bir gönderici adresi olmalıdır. Lütfen [SERVER_EMAIL](#) ve [DEFAULT_FROM_EMAIL](#) değerlerini ortamınıza uygun şekilde yapılandırın. Örneğin:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Not: Weblate tarafından e-posta gönderilmesini devre dışı bırakmak için [EMAIL_BACKEND](#) seçeneğini `django.core.mail.backends.dummy.EmailBackend` olarak ayarlayın.

Bu yapılandırma, kayıt ve parola sıfırlama e-postaları ile birlikte *tüm* e-posta gönderimini devre dışı bırakır.

Ayrıca bakınız:

Yapılandırmayı ayarlama, *Giden e-postayı yapılandırmak*, [EMAIL_BACKEND](#), [DEFAULT_FROM_EMAIL](#), [SERVER_EMAIL](#)

Allowed hosts kurulumu

Django, sitenizin sunmasına izin verilen etki alanı adlarının listesini [ALLOWED_HOSTS](#) seçeneğinde tutar. Bu seçenek boş bırakılırsa tüm istekler engellenir.

Bu seçenek, HTTP sunucunuzla eşleşecek şekilde yapılandırılmamışsa, Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS. gibi hata iletileri görürsünüz

İpucu: Docker kapsayıcısında [WEBLATE_ALLOWED_HOSTS](#) seçeneği kullanılabilir.

Ayrıca bakınız:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Doğru site etki alanını ayarlayın*

Django gizli anahtarı

`SECRET_KEY` ayarı Django tarafından çerezleri imzalamak için kullanılır. Örnek kurulumdaki değeri kullanmayıp kendiniz için oluşturacağınız gerçek bir değeri kullanmanız gerekir.

Weblate içindeki `weblate-generate-secret-key` komutunu kullanarak yeni bir anahtar oluşturabilirsiniz.

Ayrıca bakınız:

`SECRET_KEY`

Bakım görevlerini yürütmek

En iyi başarımlar için, bazı bakım görevlerini arka planda çalıştırmak iyi bir fikirdir. Bu işlemler *Celery ile arka plan görevlerini kullanmak* tarafından kendiliğinden yapılır ve şu görevleri kapsar:

- Yapılandırma sağlığı denetimi (saatlik).
- Bekleyen değişiklikleri işlemek (saatlik). Bilgi almak için: *Lazy commit işlemleri* ve `commit_pending`.
- Bileşen uyarılarını güncellemek (günlük).
- Uzak dalları güncellemek (gecelik). Bilgi almak için: `:setting: 'AUTO_UPDATE'`.
- Çeviri belleğinin JSON yedeğini almak (günlük). Bilgi almak için: *dump_memory*.
- Tam metin ve veri tabanı bakım görevleri (günlük ve haftalık görevler). Bilgi almak için: *cleanuptrans*.

3.2 sürümünde değişti: 3.2 sürümünden bu yana, bu görevleri yapmak için varsayılan olarak Celery kullanılır. Weblate içinde buna uygun yapılandırma hazırdır. Bilgi almak için: ref: *celery*.

Sistem yerel ayarları ve kodlama

Sistem yerel ayarları UTF-8 destekleyenler ile yapılandırılmalıdır. Çoğu Linux dağıtımında varsayılan ayar böyledir. Sisteminizde durumun böyle olmaması durumunda, lütfen yerel ayarları UTF-8 çeşidi olarak değiştirin.

Örneğin, `/etc/default/locale` dosyasını düzenleyerek `LANG = "C.UTF-8"` olarak ayarlayabilirsiniz.

Bazı durumlarda, her hizmetin ayrı yerel ayar yapılandırması vardır. Bunlar, dağıtımlara ve web sunuculara göre farklılık gösterebilir. Bu konuda bilgi almak için web sunucusu paketlerinizin belgelerine bakın.

Ubuntu üzerinde Apache `/etc/apache2/envvars` kullanır:

```
export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'
```

CentOS üzerinde Apache `/etc/sysconfig/httpd` (or `/opt/rh/httpd24/root/etc/sysconfig/httpd`) kullanır:

```
LANG='en_US.UTF-8'
```

Özel sertifika kuruluşu kullanmak

Weblate, HTTP istekleri sırasında SSL sertifikalarını doğrular. Varsayılan paketlerde güvenilmeyen bir özel sertifika kuruluşu kullanıyorsanız, sertifikasını güvenilir olarak eklemeniz gerekir.

Yeğlenen yaklaşım bunu sistem düzeyinde yapmaktır. Ayrıntılı bilgi almak için dağıtım belgelerinize bakın (örneğin debian üzerinde bu işlem, sertifika kuruluşunun sertifikasını `/usr/local/share/ca-certificates/` içine yerleştirip **update-ca-certificates** komutunu çalıştırarak yapılabilir).

Bu işlem yapıldıktan sonra, sistem araçları sertifikaya güvenir ve Git tarafından da bu sertifika kullanılır.

Python kodu için, kendi içinde gelen sertifika yerine istekleri sistem sertifika kuruluşu paketini kullanacak şekilde yapılandırmanız gerekir. Bu işlem, aşağıdaki kod parçasını `settings.py` içine ekleyerek yapılabilir (yol Debian sistemine özgüdür):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

İstemci varlıklarının sıkıştırılması

Weblate, bazı JavaScript ve CSS dosyalarıyla birlikte gelir. Başarımı artırmak için bunların bir istemciye gönderilmeden önce sıkıştırılması iyidir. Varsayılan yapılandırmada bu işlem çok az ek yük oluşturarak anında yapılır. Büyük kurulumlarda, çevrimdışı sıkıştırma kipini etkinleştirmeniz önerilir. Bunun yapılandırmada ayarlanması ve sıkıştırmanın her Weblate yükseltmesinde tetiklenmesi gerekir.

Yapılandırma, `django.conf.settings.COMPRESS_OFFLINE` seçeneği ve `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` seçeneği etkinleştirilerek kolayca ayarlanır (ikincisi zaten örnek yapılandırmalara eklenmiştir):

```
COMPRESS_OFFLINE = True
```

Her dağıtımda, dosyaları geçerli sürümle eşleşecek şekilde sıkıştırmanız gerekir:

```
weblate compress
```

İpucu: Resmi Docker kalıbında bu özellik etkinleştirilmiştir.

Ayrıca bakınız:

[Common Deployment Scenarios](#), *Durağan dosyalar sunmak*

2.1.10 Sunucuyu çalıştırmak

İpucu: Aşağıda açıklanan hizmetlerle ilgili deneyiminiz yoksa, *Docker ile kurmak* adresine bakmak isteyebilirsiniz.

Weblate çalıştırmak için birkaç hizmete gerek duyacaksınız. Önerilen kurulum şunlardan oluşur:

- Veri tabanı sunucusu (bilgi almak için: *Weblate için veri tabanı kurulumu*)
- Ön bellek sunucusu (bilgi almak için: *Ön bellek özelliğini açın*)
- Durağan dosyalar ve SSL ucu için ön yüz web sunucusu (bilgi almak için: *Durağan dosyalar sunmak*)
- Devingen içerik için WSGI sunucusu (bilgi almak için: *NGINX ve uWSGI için örnek yapılandırma*)
- Arka plan görevlerini yürütmek için Celery (bilgi almak için: *Celery ile arka plan görevlerini kullanmak*)

Not: Hizmetler arasında bazı bağımlılıklar bulunur. Örneğin Celery veya uwsgi işlemlerini başlatırken ön bellek ve veritabanı çalışıyor olmalıdır.

Çoğu durumda, tüm hizmetleri tek (sanal) sunucu üzerinde çalıştırırsınız. Ancak kurulumunuzun yükü ağırsa, hizmetleri ayırabilirsiniz. Bu durumda, Celery ve wsgi sunucularının `DATA_DIR` klasörüne erişmesini sağlamanız yerlidir.

Not: WSGI işlemini çalıştıran kullanıcı ile Celery işlemini çalıştıran kullanıcı aynı olmalıdır. Yoksa `DATA_DIR` klasöründeki dosyaların sahiplikleri karışır ve bu da çalışma sırasında sorunlara yol açar.

Ayrıca *Dosya sistemi izinleri* ve *Celery ile arka plan görevlerini kullanmak* bölümlerine bakın.

Web sunucusunu çalıştırmak

Weblate çalıştırmak, diğer Django temelli programları çalıştırmaktan farklı değildir. Django genellikle uWSGI ya da fcgi olarak çalıştırılır (aşağıda farklı web sunucuları için örnekler bulabilirsiniz).

Deneme amacıyla, Django üzerindeki yerleşik web sunucusunu kullanabilirsiniz:

```
weblate runserver
```

Uyarı: BU SUNUCUYU ÜRETİM AYARIYLA KULLANMAYIN. Güvenlik denetimlerinden ya da başarımlarından geçirilmemiştir. Ayrıca `runserver` ile ilgili Django belgelerine bakın.

İpucu: Django içindeki sunucu, yalnızca geliştirme amaçlı olduğundan yalnızca `DEBUG` seçeneği etkin olan duran dosyaları sunar. Üretim kullanımı için lütfen *NGINX ve uWSGI için örnek yapılandırma*, *Örnek Apache yapılandırması*, *Örnek Apache ve Unicorn yapılandırması* ve *Duran dosyalar sunmak* içindeki wsgi kurulumlarına bakın.

Duran dosyalar sunmak

2.4 sürümünde değişti: 2.4 sürümünden önce Weblate, Django duran dosyalar çatısını düzgün kullanmıyordu ve kurulum daha karmaşıktı.

Duran Django dosyalarının tek bir klasörde toplanması gerekiyor. Bunun için `weblate collectstatic --noinput` komutunu çalıştırın. Bu komut, duran dosyaları `STATIC_ROOT` seçeneği ile belirtilen bir klasöre kopyalar (bu klasör, varsayılan olarak `DATA_DIR` içindeki `static` klasörüdür).

Duran dosyaları doğrudan web sunucunuzdan sunmanız önerilir. Bunu şu yollar ile kullanmalısınız:

/static/

Weblate ve yönetici arayüzünün duran dosyalarını sunar (`STATIC_ROOT` ile tanımlanmış).

/media/

Kullanıcıların ortam yüklemeleri için kullanılır (ekran görüntüleri gibi).

/favicon.ico

`/static/favicon.ico` dosyasını sunmak için bir kural yeniden yazılmalıdır.

Ayrıca bakınız:

NGINX ve uWSGI için örnek yapılandırma, *Örnek Apache yapılandırması*, *Örnek Apache ve Unicorn yapılandırması*, *İstemci varlıklarının sıkıştırılması*, *How to deploy Django*, *How to deploy static files*

İçerik güvenliği ilkesi

Varsayılan Weblate yapılandırması, Content-Security-Policy veya X-XSS-Protection gibi güvenlikle ilgili HTTP üst bilgilerini ayarlayan `weblate.middleware.SecurityMiddleware` ara yazılımını etkinleştirir. Bunlar varsayılan olarak Weblate ve yapılandırması ile çalışacak şekilde ayarlanmıştır. Ancak bunun için ortamınıza uygun özelleştirmeler yapmanız gerekebilir.

Ayrıca bakınız:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

NGINX ve uWSGI için örnek yapılandırma

Üretim amacıyla kullanılacak web sunucusu çalıştırmak için, Weblate ile kurulan wsgi sarmalayıcısını kullanın (sanal ortamdaysanız `~/weblate-env/lib/python3.9/site-packages/weblate/wsgi.py` olarak kurulur). Kullandığınız virtualenv üzerinde Python arama yolunu ayarlamayı da unutmayın (uWSGI üzerinde `virtualenv = /home/user/weblate-env` gibi).

Aşağıdaki yapılandırma, NGINX web sunucusu üzerinde uWSGI olarak Weblate çalıştırır.

NGINX için yapılandırma (`weblate/examples/weblate.nginx.conf` dosyasında da bulunabilir):

```
#
# nginx configuration for Weblate
#
# You will want to change:
#
# - server_name
# - 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
#
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:
    }
}
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

uwsgi_pass unix:///run/uwsgi/app/weblate/socket;
# uwsgi_pass 127.0.0.1:8080;
}
}

```

uWSGI için yapılandırma (weblate/examples/weblate.uwsgi.ini dosyasında da bulunabilir):

```

#
# 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

# 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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

Ayrıca bakınız:

How to use Django with uWSGI

Örnek Apache yapılandırması

Weblate ile WSGI kullanırken prefork MPM kullanılması önerilir.

Aşağıdaki yapılandırma WSGI olarak Weblate çalıştırır. `mod_wsgi` modülünün etkinleştirilmiş olması gerekir (weblate/examples/apache.conf dosyasında bulunabilir):

```
#
# 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
#
<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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    </Files>
  </Directory>

</VirtualHost>

```

Not: Weblate için Python 3 gereklidir. Bu nedenle modwsgi modülünün Python 3 çeşidini çalıştırdığınızdan emin olun. Genellikle libapache2-mod-wsgi-py3 gibi ayrı bir paket olarak bulunur.

Ayrıca bakınız:

Sistem yerel ayarları ve kodlama, [How to use Django with Apache and mod_wsgi](#)

Örnek Apache ve Gunicorn yapılandırması

Aşağıdaki yapılandırma Gunicorn ve Apache 2.4 üzerinde Weblate çalıştırır (weblate/examples/apache.gunicorn.conf dosyasında bulunabilir):

```

#
# 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
# - 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/

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
ProxyPreserveHost On
</VirtualHost>
```

Ayrıca bakınız:

How to use Django with Gunicorn

Bir yol altında Weblate çalıştırma yapılandırması

1.3 sürümünde geldi.

Weblate ile WSGI kullanırken prefork MPM kullanılması önerilir.

Weblate çalıştırmak için /weblate gibi bir yol kullanan örnek Apache yapılandırması. Yine mod_wsgi kullanarak (weblate/examples/apache-path.conf dosyasında bulunabilir):

```
#
# 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>
    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>
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
</VirtualHost>
```

Ek olarak `weblate/settings.py` dosyasını ayarlamalısınız:

```
URL_PREFIX = "/weblate"
```

2.1.11 Celery ile arka plan görevlerini kullanmak

3.2 sürümünde geldi.

Weblate, düzenli olarak yapılan arka plan görevlerini yerine getirmek için Celery kullanır. Bu işlemleri yapacak bir Celery hizmeti çalıştırmanız gerekiyor. Örnek olarak, aşağıdaki işlemlerin yapılır (bu liste tam değildir):

- Dış hizmetlerden web kancalarını almak (bilgi almak için: *Bildirim kancaları*).
- Yedeklemeler, temizlemeler, günlük eklentiler veya güncellemeler gibi düzenli bakım görevlerini çalıştırmak (bilgi almak için *Weblate yedeğini alma ve taşıma*, *BACKGROUND_TASKS*, *Eklentiler*).
- *Kendiliğinden çeviri* çalıştırmak.
- Toplu bildirimleri göndermek.
- Yük oluşturan işlemleri wsgi işlemi üzerinden almak.
- Bekleyen değişiklikleri göndermek (Bilgi almak için: *Lazy commit işlemleri*).

Arka uçta Redis kullanan tipik bir kurulum şöyle görünür:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Ayrıca bakınız:

Celery içindeki Redis işleyici yapılandırması

Görevleri işlemek ve zamanlanmış görevleri başlatmak için Celery işlemini de başlatmalısınız. Bu işlem doğrudan komut satırından yapılabilir (genellikle hata ayıklama veya geliştirme sırasında yararlıdır):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Not: Celery işlemini çalıştıran kullanıcı, WSGI işlemini çalıştıran kullanıcı ile aynı olmalıdır. Yoksa *DATA_DIR* klasöründeki dosyaların sahiplikleri karışır ve bu da çalışma sırasında sorunlara yol açar.

Ayrıca *Dosya sistemi izinleri* ve *Sunucuyu çalıştırmak* bölümlerine bakabilirsiniz.

Wsgi üzerinde Celery görevlerini yürütürken eager kipini kullanmak

Not: Bu seçeneğin kullanılmasının web arabirimi başarımı üzerinde ciddi etkisi olur ve normal tetikleyiciye bağlı olarak özellikleri bozar (bekleyen değişiklikleri gerçekleştirmek, özet bildirimleri veya yedeklemeler gibi).

Geliştirme ortamları için, tüm görevleri anında işleyen eager yapılandırmasını kullanmak isteyebilirsiniz:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Celery uygulamasını sistem hizmeti olarak çalıştırmak

Büyük olasılıkla `celery:userguide/daemonizing` bölümünde anlatıldığı gibi bir Celery daemon çalıştırmak isteyeceksiniz. `systemd` kullanan en yaygın Linux kurulumu için, hazır gelen ve aşağıda listesi bulunan `examples` klasöründeki örnek dosyaları kullanabilirsiniz.

`/etc/systemd/system/celery-weblate.service` olarak yerleştirilecek `systemd` birimi:

```
[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} \
  -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
```

`/etc/default/celery-weblate` olarak yerleştirilecek ortam yapılandırması:

```
# 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"
```

Celery günlüklerini döndürmek için **logrotate** komutu kullanılarak `/etc/logrotate.d/celery` olarak yerleştirilecek ek yapılandırma:

```
/var/log/celery/*.log {  
    weekly  
    missingok  
    rotate 12  
    compress  
    notifempty  
}
```

Celery atımını kullanarak görevleri zamanlamak

Weblate, hazır bir zamanlanmış görevler kurulumuyla gelir. Bununla birlikte, `settings.py` dosyasından ek görevler tanımlayabilirsiniz. Örnek olarak *Lazy commit işlemleri* bölümüne bakabilirsiniz.

Görevlerin Celery beats daemon tarafından yürütülmesi gerekiyor. Beklendiği gibi çalışmıyorsa, daemon çalışmıyor veya veritabanı bozulmuş olabilir. Bu durumda temel sorunu bulmak için Celery başlangıç günlüklerine bakın.

Celery durumunu izlemek

Celery görev kuyruklarının geçerli uzunluğunu *Yönetim arayüzü* içinden ya da komut satırında `celery_queues` komutunu kullanarak görebilirsiniz. Kuyruğun çok uzaması durumunda, yönetici arayüzünde de yapılandırma hatası görürsünüz.

Uyarı: Celery hataları varsayılan olarak yalnızca Celery günlüğüne kaydedilir ve kullanıcı tarafından görülemez. Bu tür hataların özetini görmek istiyorsanız, *Hata raporlarını derlemek* seçeneğini yapılandırmanız önerilir.

Ayrıca bakınız:

Weblate uygulamasını izlemek, *How can I check whether my Weblate is set up properly?*, *Configuration and defaults*, *Workers Guide*, *Daemonization*, *Monitoring and Management Guide*, `celery_queues`

2.1.12 Weblate uygulamasını izlemek

Weblate, örneğin Kubernetes gibi basit durum denetimlerinde kullanılmak üzere `/healthz/` adresini sunar. Docker kapsayıcısında, bu adresi kullanan iç sistem durumu denetimi bulunur.

Weblate ölçümlerini izlemek için `GET /api/metrics/` API uç noktasını kullanabilirsiniz.

Ayrıca bakınız:

How can I check whether my Weblate is set up properly?, *Celery durumunu izlemek*, *Munin için Weblate uygulama eki*

2.1.13 Hata raporlarını derlemek

Weblate, diğer yazılımlar gibi sorun çıkarabilir. Yardımcı olabilecek sorun durumlarını derlemek için üçüncü taraf hizmetlerini kullanmanızı öneririz. Bu uygulama, özellikle Celery görevlerinin yapılamaması durumunda kullanışlıdır. Yoksa hata yalnızca günlüklere bildirilir ve bunlar hakkında bildirim almazsınız. Weblate tarafından desteklenen hizmetler şunlardır:

Sentry

Weblate, [Sentry](#) desteği sunar. Kullanmak için, `settings.py` dosyasında `SENTRY_DSN` seçeneğini ayarlamak yeterlidir:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate, [Rollbar](#) desteği sunar. Kullanmak için, [Rollbar notifier for Python](#) yönergelerini izlemek yeterlidir.

Özetle, `settings.py` dosyasını ayarlamanız gerekir:

```
# 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",
}
```

Bunun dışındaki her şey kendiliğinden bütünleştirilmiştir. Artık hem sunucu hem de istemci tarafı hatalarını derleyebilirsiniz.

2.1.14 Weblate kurulumunu başka bir sunucuya aktarmak

Weblate kurulumunu başka bir sunucuya aktarmak oldukça kolaydır. Ancak veriler dikkatli bir şekilde aktarmanız gereken birkaç konumda bulunur. En iyi yaklaşım, aktarım sırasında Weblate kopyasını durdurmaktır.

Veri tabanını aktarmak

Veritabanı arka ucunuza bağlı olarak, veritabanını aktarmak için birkaç seçeneğiniz olabilir. En basit yaklaşım, genellikle en etkili araçlar olduklarından, veritabanının kendi araçlarını kullanmaktır (**`mysqldump`** ya da **`pg_dump`** gibi). Alternatif olarak, veritabanınızın desteklemesi durumunda çoğaltma (replication) özelliğini kullanabilirsiniz.

Ayrıca bakınız:

Veri tabanları arasında aktarım *[Diğer veri tabanlarından PostgreSQL üzerine aktarım](#)* bölümünde açıklanmıştır.

Sürüm denetimi sistemi depolarını aktarmak

`DATA_DIR` altına kaydedilmiş sürüm denetimi sistemi depolarının da taşınması gerekir. Aktarımı daha etkili bir şekilde yapmak için bunları kopyalayabilir ya da **`rsync`** komutunu kullanabilirsiniz.

Diğer notlar

Weblate tarafından kullanılıyor olabilecek, Redis, zamanlanmış Cron görevleri veya özel kimlik doğrulama arka uçları gibi diğer hizmetleri taşımayı unutmayın.

2.2 Weblate dağıtımları

Weblate, bulutunuza kolayca kurulabilir. Lütfen platformunuz için ayrıntılı bilgiler içeren rehbera bakın:

- *Docker ile kurmak*
- *OpenShift üzerine kurulum*
- *Kubernetes üzerine kurulum*

2.2.1 Üçüncü taraf Weblate dağıtımları

Not: Şu dağıtımlar Weblate ekibi tarafından geliştirilmez ya da desteklenmez. Kurulum aşamaları bu belgede anlaşılan farklı olabilir.

Bitnami Weblate paketi

Bitnami, <<https://bitnami.com/stack/weblate>> adresinden birçok platform için hazır Weblate paketleri sunar.

Ayrıca bakınız:

[Bitnami Weblate paketi](#)

Cloudron Weblate paketi

Cloudron web uygulamalarını kendinizin barındırabileceğiniz bir platformdur. Cloudron ile kurulan Weblate kendiliğinden güncel tutulur. Paket, Cloudron ekibi tarafından [Weblate paket deposu](#) üzerinde güncel tutulur.



YunoHost üzerinde Weblate

Kendi kendine barındırma projesi [YunoHost](#) bir Weblate paketi sağlar. YunoHost kurulumunuzu yaptıktan sonra, Weblate başka bir uygulama olarak kurulabilir. Yedekleme ve geri yükleme ile tam olarak çalışan bir paket sunar. Bununla birlikte yine de belirli kullanımlar için ayarlar dosyanızı düzenlemeniz gerekebilir.

Yönetim arayüzünü ya da bu düğmeyi kullanabilirsiniz (sizi sunucunuza götürür):



Komut satırı arayüzü de kullanılabilir:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```


2.3 Upgrading Weblate

2.3.1 Docker kalıbı güncellemeleri

The official Docker image (see [Docker ile kurmak](#)) has all Weblate upgrade steps integrated. There are typically no manual steps needed besides pulling latest version.

Ayrıca bakınız:

[Docker kapsayıcısını yükseltmek](#)

2.3.2 Generic upgrade instructions

Before upgrading, please check the current [Yazılım gereksinimleri](#) 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 [Sürüme özel yönergeler](#) 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.

Not: It is recommended to perform a full database backup prior to upgrade so that you can roll back the database in case upgrade fails, see [Weblate yedeğini alma ve taşıma](#).

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. Weblate kodunu yükseltin.

Pip kurulumları için şöyle yapılabilir:

```
pip install -U "Weblate[all] ==version"
```

Or, if you just want to get the latest released version:

```
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 [İsteğe bağlı bağımlılıklar](#), please check if they cover features you want.
4. Upgrade configuration file, refer to `settings_example.py` or [Sürüme özel yönergeler](#) for needed steps.
5. Veri tabanı yapısını yükseltin:

```
weblate migrate --noinput
```

6. Collect updated static files (see *Sunucuyu çalıştırmak* and *Durağan dosyalar sunmak*):

```
weblate collectstatic --noinput --clear
```

7. Compress JavaScript and CSS files (optional, see *İstemci varlıklarının sıkıştırılması*):

```
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. Kurulumunuzun doğru olduğunu doğrulayın (bilgi almak için: `:ref:production`):

```
weblate check --deploy
```

10. Celery işlemini yeniden başlatın (bilgi almak için: *Celery ile arka plan görevlerini kullanmak*).

2.3.3 Sürüme özel yönergeler

2.x sürümünden yükseltmek

2.x sürümünden yükseltiyorsanız, her zaman ilk olarak 3.0.1 sürümüne yükseltin. Ardından 3.x serisinde yükseltme işlemini sürdürün. Bu adımı atlayan yükseltme işlemleri desteklenmez ve bozulur.

Ayrıca bakınız:

Weblate 3.0 belgelerinde 2.20 sürümünden 3.0 sürümüne yükseltme

3.x sürümünden yükseltmek

3.x sürümünden yükseltme yapıyorsanız, her zaman ilk olarak 4.0.4 veya 4.1.1 sürümüne yükseltin. Ardından 4.x serisinde yükseltmeyi sürdürün. Bu adımı atlayan yükseltmeler desteklenmez ve bozulur.

Ayrıca bakınız:

Weblate 4.0 belgelerinde 3.11 sürümünden 4.0 sürümüne yükseltmek

4.0 sürümünden 4.1 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- 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.
- Bazı yeni ve güncellenmiş gereksinimler vardır.
- `INSTALLED_APPS` içinde bir değişiklik var.

- The `MT_DEEPL_API_VERSION` setting has been removed in Version 4.7. The *DeepL* machine translation now uses the new `MT_DEEPL_API_URL` instead. You might need to adjust `MT_DEEPL_API_URL` to match your subscription.

Ayrıca bakınız:

Generic upgrade instructions

4.1 sürümünden 4.2 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- Upgrade from 3.x releases is not longer supported, please upgrade to 4.0 or 4.1 first.
- Bazı yeni ve güncellenmiş gereksinimler vardır.
- 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.

Ayrıca bakınız:

Generic upgrade instructions

4.2 sürümünden 4.3 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- 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 istemcisi* 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).
- `INSTALLED_APPS` içinde bir değişiklik var.
- 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`.

4.3.1 sürümünde değişti:

- The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

4.3.2 sürümünde değişti:

- The `post_update` method of add-ons now takes extra `skip_push` parameter.

Ayrıca bakınız:

Generic upgrade instructions

4.3 sürümünden 4.4 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- There is a change in `INSTALLED_APPS`, `weblate.configuration` has to be added there.
- Artık Django 3.1 sürümü gerekiyor.
- In case you are using MySQL or MariaDB, the minimal required versions have increased, see *MySQL ve MariaDB*.

4.4.1 sürümünde değişti:

- *Tek dilli 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.

Ayrıca bakınız:

Generic upgrade instructions

4.4 sürümünden 4.5 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- The migration might take considerable time if you had big glossaries.
- Sözlükler artık normal bileşenler olarak tutuluyor.
- The glossary API is removed, use regular translation API to access glossaries.
- There is a change in `INSTALLED_APPS` - `weblate.metrics` should be added.

4.5.1 sürümünde değişti:

- There is a new dependency on the *pyahocorasick* module.

Ayrıca bakınız:

Generic upgrade instructions

4.5 sürümünden 4.6 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- 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 *İç Weblate adresleri*, see `POST /api/projects/(string:project)/components/`.
- There is a change in dependencies and `PASSWORD_HASHERS` to prefer Argon2 for passwords hashing.

Ayrıca bakınız:

Generic upgrade instructions

4.6 sürümünden 4.7 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

Önemli yapılandırma veya bağımlılık değişiklikleri:

- There are several changes in `settings_example.py`, most notable middleware changes (`MIDDLEWARE`), please adjust your settings accordingly.
- The *DeepL* machine translation now has a generic `MT_DEEPL_API_URL` setting to adapt to different subscription models more flexibly. The `MT_DEEPL_API_VERSION` setting is no longer used.
- Artık Django 3.2 sürümü gerekiyor.

Ayrıca bakınız:

Generic upgrade instructions

4.7 sürümünden 4.8 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

There are no additional upgrade steps needed in this release.

Ayrıca bakınız:

Generic upgrade instructions

4.8 sürümünden 4.9 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- There is a change in storing metrics, the upgrade can take long time on larger sites.

Ayrıca bakınız:

Generic upgrade instructions

4.9 sürümünden 4.10 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- 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:
 - Dropped support for PostgreSQL 9.6, Django 4.0 supports PostgreSQL 10 and higher.
 - Format of `CSRF_TRUSTED_ORIGINS` was changed.
- The Docker container now uses Django 4.0, see above for changes.

Ayrıca bakınız:

Generic upgrade instructions

4.10 sürümünden 4.11 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- Weblate için artık Django 3.2 gerekiyor.
- The implementation of *Her proje için erişim denetimi yönetimi* has changed, removing the project prefix from the group names. This affects API users.
- Weblate now uses `charset-normalizer` instead of `chardet` module for character set detection.
- **Changed in 4.11.1:** There is a change in `REST_FRAMEWORK` setting (removal of one of the backends in `DEFAULT_AUTHENTICATION_CLASSES`).

Ayrıca bakınız:

Generic upgrade instructions

4.11 sürümünden 4.12 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- Herhangi bir özel adım gerekmez.

Ayrıca bakınız:

Generic upgrade instructions

4.12 sürümünden 4.13 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- The *Dil tanımları* are now automatically updated on upgrade, use `UPDATE_LANGUAGES` to disable that.
- Handling of context and location has been changed for *Windows RC dosyaları*, *HTML dosyaları*, *IDML biçimi*, and *Metin dosyaları* file formats. In most cases the context is now shown as location.
- The machine translation services are now configured using the user interface, settings from the configuration file will be imported during the database migration.

Ayrıca bakınız:

Generic upgrade instructions

4.13 sürümünden 4.14 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- The Java formatting checks now match GNU gettext flags. The flags set in Weblate will be automatically migrated, but third-party scripts will need to use `java-printf-format` instead of `java-format` and `java-format` instead of `java-messageformat`.
- The *jellyfish* dependency has been replaced by *rapidfuzz*.
- **4.14.2 sürümünde değiştirildi:** VCS hizmeti API anahtarlarının güvenli olmayan yapılandırması, `_CREDENTIALS` listesi yerine `_TOKEN/_USERNAME/` yapılandırmasıyla kullanımdan kaldırıldı. Docker üzerinde lütfen eşleşen `_HOST` yönergesini ekleyin. Örneğin `WEBLATE_GITHUB_HOST` ve `GIT-HUB_CREDENTIALS` bölümlerine bakın.

Ayrıca bakınız:

Generic upgrade instructions

4.14 sürümünden 4.15 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- Weblate now requires `btree_gin` extension in PostgreSQL. The migration process will install it if it has sufficient privileges. See *PostgreSQL üzerinde bir veri tabanı oluşturmak* for manual setup.
- The Docker image no longer enables debug mode by default. In case you want it, enable it in the environment using `WEBLATE_DEBUG`.
- The database migration may take hours on larger instances due to recreating some of the indexes.
- **Changed in 4.15.1:** The default value for `DEFAULT_PAGINATION_CLASS` in rest framework settings was changed.

Ayrıca bakınız:

Generic upgrade instructions

4.15 sürümünden 4.16 sürümüne yükseltmek

Güncellemek için *Generic upgrade instructions* adımlarını izleyin.

- 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 *4.13 sürümünden 4.14 sürümüne yükseltmek*.
- *django-crispy-forms* yükseltmesi için `INSTALLED_APPS` içinde değişiklikler yapılmalıdır.
- Integration of *django-cors-headers* requires changes in `INSTALLED_APPS` and `MIDDLEWARE`.

Ayrıca bakınız:

Generic upgrade instructions

2.3.4 Python 2 sürümünden Python 3 sürümüne yükseltmek

Weblate no longer supports Python older than 3.6. In case you are still running on older version, please perform migration to Python 3 first on existing version and upgrade later. See *Upgrading from Python 2 to Python 3 in the Weblate 3.11.1 documentation*.

2.3.5 Diğer veri tabanlarından PostgreSQL üzerine aktarım

If you are running Weblate on other database than PostgreSQL, you should consider migrating to PostgreSQL as Weblate performs best with it. The following steps will guide you in migrating your data between the databases. Please remember to stop both web and Celery servers prior to the migration, otherwise you might end up with inconsistent data.

PostgreSQL üzerinde bir veri tabanı oluşturmak

Weblate için ayrı bir kullanıcı hesabı ile ayrı bir veri tabanı kullanmak genellikle iyi bir fikirdir:

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Django JSON dökümlerini kullanarak aktarım

The simplest approach for migration is to utilize Django JSON dumps. This works well for smaller installations. On bigger sites you might want to use pgloader instead, see [Migrating to PostgreSQL using pgloader](#).

1. Add PostgreSQL as additional database connection to the `settings.py`:

```
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. Run migrations and drop any data inserted into the tables:

```
weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql
```

3. Dump legacy database and import to PostgreSQL


```
weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql
```

4. Adjust `DATABASES` to use just PostgreSQL database as default, remove legacy connection.

Weblate should be now ready to run from the PostgreSQL database.

Migrating to PostgreSQL using pgloader

The `pgloader` is a generic migration tool to migrate data to PostgreSQL. You can use it to migrate Weblate database.

1. Adjust your `settings.py` to use PostgreSQL as a database.
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 Pootle üzerinden aktarım

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 Weblate yedeğini alma ve taşıma

2.4.1 Proje düzeyinde yedekler

4.14 sürümünde geldi.

Uyarı: Yedeklemeler yalnızca veri tabanı olarak PostgreSQL ya da MariaDB 10.5+ kullanıldığında geri yüklenebilir.

Proje, Weblate üzerindeki tüm çeviri içeriğini yedekler (proje, bileşenler, çeviriler, dizge yorumları, öneriler ve denetimler). Bir projeyi başka bir Weblate kopyasına aktarmak için uygundur.

Bir projeyi *Yönetim* ↓ *Yedekler* bölümünden yedekleyebilirsiniz. Bir proje oluşturulurken yedekleme geri yüklenebilir (bilgi almak için: *Adding translation projects and components*).

Şu anda yedeklerde erişim denetimi ve geçmiş bilgileri bulunmuyor.

Yorumlar ve öneriler, bunları oluşturan kullanıcının kullanıcı adıyla yedeklenir. İçer aktarıldıktan sonra eşleşen bir kullanıcı ile ilişkilendirilir. Yedekteki kullanıcı adına sahip bir kullanıcı bulunamazsa, anonim kullanıcı ile ilişkilendirilir.

Oluşturulan yedekler, `PROJECT_BACKUP_KEEP_DAYS` ve `PROJECT_BACKUP_KEEP_COUNT` tarafından yapılandırıldığı şekilde sunucuda tutulur (varsayılan olarak 30 gün boyunca en fazla 3 yedek tutulur).

2.4.2 BorgBackup ile kendiliğinden yedekleme

3.9 sürümünde geldi.

Weblate, **BorgBackup** kullanarak hizmet yedeklerinin alınmasını destekler. Borg, bulutta güvenli bir şekilde saklanabilen, depolama alanı açısından etkili şifrelenmiş yedekler alır. Yedekler, yönetim arayüzünde *Yedekler* sekmesinden yönetilebilir.

4.4.1 sürümünde değişti: Kendiliğinden alınan yedeklere hem PostgreSQL hem de MySQL/MariaDB veri tabanları katılır.

Borg kullanan yedekler artımlıdır ve Weblate şu yedekleri koruyacak şekilde yapılandırılır:

- 14 gün geriye dönük günlük yedekler
- 8 hafta geriye dönük haftalık yedekler
- 6 ay geriye dönük aylık yedekler

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 şifreleme anahtarı

BorgBackup şifrelenmiş yedekler alır ve parola olmadan bunları geri yükleyemezsiniz. Parola, yeni bir yedekleme hizmeti eklerken oluşturulur ve bunu kopyalayıp güvenli bir yerde saklamanız gerekir.

Weblate tarafından sunulan yedekleme depolama alanı kullanıyorsanız, yedeklerinize erişmek için kullanıldığından, lütfen kişisel SSH anahtarınızı da yedekleyin.

Ayrıca bakınız:

`borg init`

Yedeklemeyi özelleştirme

- Veri tabanı yedeklemesi `DATABASE_BACKUP` ile yapılandırılabilir.
- Yedek oluşturma `BORG_EXTRA_ARGS` kullanılarak özelleştirilebilir.

2.4.3 Weblate tarafından sunulan yedekleme depolama alanı

Weblate bulut sunucunuzu yedeklemenin en kolay yolu **`weblate.org <https://weblate.org/support/#backup> üzerinden yedekleme hizmeti`** satın almaktır. Şu şekilde çalışır duruma getirebilirsiniz:

1. <https://weblate.org/support/#backup> üzerinden *Yedekleme hizmeti* satın alın.
2. Size verilen anahtarı yönetim bölümüne yazın. Bilgi almak için: *Destek bütünleşmesi*.
3. Weblate bulut hizmeti ile bağlantı kurar ve yedekler için erişim bilgilerini alır.
4. *Yedekler* sekmesinden yeni yedekleme yapılandırmasını etkinleştirin.
5. Yedekleri geri yükleyebilmek için Borg kimlik doğrulama bilgilerinizi yedekleyin. Bilgi almak için: *Borg şifreleme anahtarı*.

İpucu: Güvenliğiniz için el ile her şeyi etkinleştirme adımı bulunur. İzniniz olmadan, hesap açma işlemi yoluyla elde edilen yedekleme havuzuna herhangi bir veri gönderilmez.

2.4.4 Özel yedekleme depolama alanını kullanmak

Yedekler için kendi depolama alanınızı da kullanabilirsiniz. SSH, yedekleri uzak hedefte depolamak için kullanılabilir. Hedef sunucuda *BorgBackup* kurulu olmalıdır.

Ayrıca bakınız:

Borg belgelerindeki *General* bölümünde

Yerel dosya sistemi

Yerel yedekleme için mutlak yolun belirtilmesi önerilir. Örneğin `/yedek/klasoru/yolu`. Klasör, Weblate çalıştıran kullanıcı tarafından yazılabilir olmalıdır (bilgi almak için: *Dosya sistemi izinleri*). Klasör yoksa, Weblate oluşturmaya çalışır, ancak bunu yapmak için uygun izinlere gerek duyar.

İpucu: Docker üzerinde Weblate çalıştırırken, lütfen yedekleme konumunun Weblate kapsayıcısında bir birim olarak gösterildiğinden emin olun. Yoksa, içinde bulunduğu kapsayıcı yeniden başlatıldığında yedekler Docker tarafından atılır.

Yedekleri `/app/data/borgbackup` gibi var olan bir birime yerleştirmek bir seçenek olabilir. Bu birim kapsayıcıda vardır.

Docker Compose dosyasındaki yedekler için örneğin `/borgbackup` kullanarak yeni bir kapsayıcı da ekleyebilirsiniz:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

Yedeklerin depolanacağı klasör için UID 1000 olmalıdır. Yoksa Weblate yedekleri bu klasöre yazamaz.

Uzak yedeklemeler

Uzak yedekler almak için, Weblate SSH anahtarını kullanarak SSH üzerinden Weblate dağıtımınız için erişilebilen başka bir sunucuya **BorgBackup** kurmanız gerekir:

1. Yedeklerinizin depolanacağı bir sunucu hazırlayın.
2. SSH sunucusunu üzerine kurun (çoğu Linux dağıtımında varsayılan olarak bulunur).
3. Bu sunucuya **BorgBackup** kurun. Çoğu Linux dağıtımında kullanılabilir paketleri bulunur (bilgi almak için: [Installation](#)).
4. Var olan bir kullanıcı seçin ya da yedekleme için kullanılacak yeni bir kullanıcı ekleyin.
5. Weblate tarafından parola olmadan sunucu ile SSH bağlantısının kurulabilmesi için kullanıcıya Weblate SSH anahtarını ekleyin (bilgi almak için: [Weblate SSH anahtarı](#)).
6. Yerel yedeklemeler için `kullanici@sunucu:/yedeklere/giden/yol` şeklinde ya da uzak SSH yedeklemeleri için `ssh://kullanici@sunucu:baglantinoktasi/yedeklere/giden/yol` şeklinde kullanın.

İpucu: *Weblate tarafından sunulan yedekleme depolama alanı* herhangi bir çaba harcamadan kendiliğinden uzak yedekleme olanağı sağlar.

Ayrıca bakınız:

[Weblate SSH anahtarı](#), [General](#)

2.4.5 BorgBackup yedeklerini geri yüklemek

1. Yedekleme deponuza erişimi geri yükleyin ve yedekleme parolanızı hazırlayın.
2. Sunucudaki tüm yedekleri `borg list REPOSITORY` komutunu kullanarak listeleyin.
3. İstedığınız yedeği geçerli klasöre geri yüklemek için `borg extract REPOSITORY::ARCHIVE` komutunu kullanın.
4. Veri tabanını, Weblate data klasörü içindeki backup klasörüne kaydedilen SQL dökümünden geri yükleyin (bilgi almak için: [Yedeklemeler için dökümü yapılan veriler](#)).
5. Weblate yapılandırmasını (`backups/settings.py`, bilgi almak için: [Yedeklemeler için dökümü yapılan veriler](#)) doğru konuma kopyalayın. Bilgi almak için: [Yapılandırmayı ayarlama](#).

Docker kapsayıcısını kullanırken, ayarlar dosyası zaten kapsayıcı içine eklenmiştir ve özgün ortam değişkenlerini geri yüklemeniz gerekir. `environment.yml` dosyası bu konuda size yardımcı olabilir (bilgi almak için: [Yedeklemeler için dökümü yapılan veriler](#)).

6. Geri yüklenen tüm data klasörünü `DATA_DIR` ile yapılandırılmış konuma kopyalayın.

Docker kapsayıcısını kullanırken verileri veri birimine yerleştirin. Bilgi almak için: [Docker kapsayıcısı birimleri](#).

Lütfen dosyaların sahiplik ve erişim izinlerinin doğru olduğundan emin olun. Bilgi almak için: [Dosya sistemi izinleri](#).

Borg oturumu şunun gibi görünmelidir:

```
$ 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:
```

Ayrıca bakınız:

[borg list](#), [borg extract](#)

2.4.6 El ile yedekleme

Neyi kaydetmek istediğinize bağlı olarak, Weblate tarafından her biri ilgili yere kaydedilmiş olan veri türünü yedekleyin.

İpucu: El ile yedekleme yapıyorsanız, Weblate yedekleme eksikliği uyarısını, `settings.py` içindeki `SILENCED_SYSTEM_CHECKS` ya da Docker için `WEBLATE_SILENCED_SYSTEM_CHECKS` seçeneğini `weblate.I028` ile ekleyerek kapatmak isteyebilirsiniz.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

Veri tabanı

Gerçek depolama konumu, veri tabanı kurulumunuza bağlıdır.

İpucu: Veri tabanı en önemli depolama alanıdır. Veri tabanınızı düzenli olarak yedeklenecek biçimde ayarlayın. Veri tabanı olmadan, tüm çeviriler kaybolur.

Doğal veri tabanı yedeklemesi

Veri tabanı dökümününün `pg_dump` ya da `mysqldump` gibi veri tabanına özgü araçlar kullanılarak kaydedilmesi önerilir. Başarımı genellikle Django yedeklemesinden daha iyidir ve tüm tablolar tüm verileriyle geri yüklenir.

Bu yedeği daha yeni bir Weblate sürümünde geri yükleyebilirsiniz, `migrate` içinde çalışırken gerekli tüm aktarımlar yapılır. Sürümler arasında nasıl yükseltme yapılacağı hakkında bilgi almak için [Upgrading Weblate](#) bölümüne bakabilirsiniz.

Django veri tabanı yedeklemesi

Alternatif olarak, `dumpdata` Django komutunu kullanarak veri tabanınızı yedekleyebilirsiniz. Bu şekilde, alınan yedek veri tabanından bağımsız olur ve veri tabanı işleyicisini değiştirmek istediğinizde kullanılabilir.

Veri tabanını geri yüklemekten önce, yedeklemenin yapıldığı Weblate sürümüyle tam olarak aynı sürümü çalıştırıyor olmanız gerekir. Veri tabanı yapısı sürümler arasında değiştiğinden ve verilerin bir şekilde bozulmasına neden olduğundan böyle olması gereklidir. Aynı sürümü kurduktan sonra, tüm veri tabanı aktarımlarını yapmak için `migrate` komutunu kullanın.

Daha sonra veri tabanında bazı kayıtlar zaten oluşturulmuş olacak ve bunlar veri tabanı yedeğinde de bulunuyor olacak. Önerilen yaklaşım, yönetim kabuğunu kullanarak bu tür kayıtları el ile silmektir (bilgi almak için [Invoking management commands](#)):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Dosyalar

Yeterli yedekleme alanınız varsa, `DATA_DIR` klasörünü tümüyle yedekleyin. Bu yedek, istemediğiniz bazı dosyaları içerse bile en güvenlisidir. Aşağıdaki bölümlerde, neleri yedeklemeniz gerektiğini ve neleri atlayabileceğinizi ayrıntılı olarak görebilirsiniz.

Yedeklemeler için dökümü yapılan veriler

4.7 sürümünde değişti: Docker ortamlarında geri yüklemeye yardımcı olacak ortam dökümü `environment.yml` olarak eklenmiştir.

`DATA_DIR/backups` içine kaydedilir.

Weblate buraya çeşitli verileri döker. Daha eksiksiz yedeklemeler için bu dosyaları ekleyebilirsiniz. Dosyalar günlük olarak güncellenir (çalışan bir Çeler beats sunucusu gerekir. Bilgi almak için: [Celery ile arka plan görevlerini kullanmak](#)). Şu anda, bulunan veriler:

- Weblate ayarları için `settings.py` (`settings-expanded.py` dosyasında genişletilmiş bir sürümü de bulunur).
- PostgreSQL veri tabanı yedeği için `database.sql`.
- Ortam dökümü için `environment.yml`.

Veri tabanı yedekleri varsayılan olarak düz metin biçiminde kaydedilir. Ancak `DATABASE_BACKUP` seçeneği ile sıkıştırılabilir ya da tümüyle atlanabilir.

Veri tabanı yedeğini geri yüklemek için veri tabanı araçlarını kullanın. Örneğin:

```
psql --file=database.sql weblate
```

Sürüm denetimi depoları

`DATA_DIR/vcs` konumunda tutulur.

Sürüm denetimi depolarında, Weblate değişiklikleriyle birlikte yukarı akış depolarınızın bir kopyası bulunur. Tüm çeviri bileşenleriniz için *İşleme ile itme* seçeneği etkinse, tüm Weblate değişiklikleri yukarı akışa dahil edilir. Weblate tarafındaki depoları yedeklemek gerekmez çünkü veri kaybı olmadan yukarı akış konumlarından yeniden kopyalanabilir.

SSH ve GPG anahtarları

`DATA_DIR/ssh` ve `DATA_DIR/home` içinde bulunur.

Weblate tarafından oluşturulmuş SSH ya da GPG anahtarlarını kullanıyorsanız, bu konumları yedeklemeniz gerekir. Yoksa kişisel anahtarlarınızı kaybedersiniz ve yeniden oluşturmanız gerekir.

Kullanıcının yüklediği dosyalar

`DATA_DIR` /media konumunda tutulur.

Kullanıcı tarafından yüklenen tüm dosyaları yedeklemelisiniz (*Visual context for strings* gibi).

Celery görevleri

Celery görev kuyruğunda bazı bilgiler bulunabilir. Ancak genellikle yedeklenmesi gerekli değildir. En fazla, henüz çeviri belleğine işlenmemiş güncellemeleri kaybedersiniz. Gene de, tam metin ya da depo güncellemesinin geri yük-
lendikten sonra yapılması önerilir, bu nedenle bunların kaybedilmesi bir sorun oluşturmaz.

Ayrıca bakınız:

Celery ile arka plan görevlerini kullanmak

Komut satırından el ile yedek alma

Bir zamanlanmış görev (cron) kullanarak, her gün yürütülecek bir Bash komutu ayarlayabilirsiniz. Örneğin:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz \
↳backups vcs ssh home media fonts secret
```

`XZ_OPT` işleminden sonra tırnak işaretleri arasındaki dizge, xz seçeneklerinizi, örneğin sıkıştırma için kullanılan bellek miktarını seçmenizi sağlar. Bilgi almak için: <https://linux.die.net/man/1/xz>

Klasör ve dosya listesini gereksinimlerinize göre ayarlayabilirsiniz. Çeviri belleğini (yedeklemeler klasöründe) kay-
detmekten kaçınmak için şunu kullanabilirsiniz:

```
$ 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 Yedeği el ile geri yüklemek

1. Yedeklediğiniz tüm verileri geri yükleyin.
2. `:d jadmin:updategit` komutunu kullanarak tüm depoları güncelleyin.

```
weblate updategit --all
```

2.4.8 Bir Weblate kurulumunu taşımak

Yukarıdaki yedekleme ve geri yükleme yönergelerini izleyerek kurulumunuzu farklı bir sisteme taşıyın.

Ayrıca bakınız:

Python 2 sürümünden Python 3 sürümüne yükseltmek, Diğer veri tabanlarından PostgreSQL üzerine aktarım

2.5 Kimlik doğrulama

2.5.1 Kullanıcı hesabı açma

Weblate varsayılan kurulumda, yeni açılan kullanıcıların hesaplarını işlemek için web sitesindeki bir form olan python-social-auth kullanır. Yeni bir kullanıcı, e-posta adresini onayladıktan sonra, üçüncü taraf hizmetlerden birini kullanarak katkıda bulunabilir veya kimliğini doğrulayabilir.

Yeni kullanıcı hesaplarının açılmasını `REGISTRATION_OPEN` komutu ile de kapatabilirsiniz.

Kimlik doğrulama girişimleri *Hız sınırlama* değeriyle sınırlanır.

2.5.2 Kimlik doğrulama arka plan uygulamaları

Hazır Django çözümü, çeşitli sosyal ağ seçenekleri ile birlikte kimlik doğrulama için kullanılır. Bu özelliği kullanmak, diğer Django temelli projelerin kullanıcı veri tabanını içe aktarabilmenizi sağlar (bilgi almak için: [Pootle üzerinden aktarım](#)).

Django başka yollarla da kimlik doğrulaması yapacak şekilde ayarlanabilir.

Ayrıca bakınız:

Kimlik doğrulama ayarları, belgesinde resmi Docker kalıbında kimlik doğrulamasının nasıl yapılandırılacağı anlatılmıştır.

2.5.3 Sosyal ağ kimlik doğrulaması

Weblate [Welcome to Python Social Auth's documentation!](#) kullanarak, GitLab, Ubuntu, Fedora gibi birçok üçüncü taraf hizmeti ile kimlik doğrulamayı destekler.

Django Framework belgesinden genel yapılandırma yönergelerine bakabilirsiniz.

Not: Weblate varsayılan olarak, doğrulanmış bir e-posta adresi sağlamak için üçüncü taraf kimlik doğrulama hizmetlerine güvenir. Kullanmak istediğiniz hizmetlerden bazıları bunu desteklemiyorsa, lütfen bunlar için FORCE_EMAIL_VALIDATION yapılandırması ile Weblate tarafında e-posta doğrulamasını zorunlu kılın. Örneğin:

```
SOCIAL_AUTH_OPENSEUSE_FORCE_EMAIL_VALIDATION = True
```

Ayrıca bakınız:

Pipeline

Tek tek arka uçları etkinleştirmek oldukça kolaydır. Bunun için yalnızca `AUTHENTICATION_BACKENDS` seçeneği ile bir kayıt ve olabilecek belirli bir kimlik doğrulama yöntemi için gerekli anahtarları eklemek gerekir. Bazı arka uçların varsayılan olarak kullanıcı e-posta adresi sağlamadığını, açıkça istemeniz gerektiğini, yoksa kullanıcılarının yaptığı katkıların Weblate tarafından düzgün bir şekilde belirlenemeyeceğini lütfen unutmayın.

İpucu: Kimlik doğrulama arka uçlarının çoğu için HTTPS gerekir. Web sunucunuzda HTTPS özelliğini etkinleştirdikten sonra lütfen Weblate için `ENABLE_HTTPS` ya da Docker kapsayıcısı için `WEBLATE_ENABLE_HTTPS` seçeneklerini yapılandırarak düzgün bir şekilde bildirilmesini sağlayın.

Ayrıca bakınız:

Python Social Auth arka yüzü

OpenID kimlik doğrulaması

OpenID kullanan hizmetler için genellikle yalnızca onları etkinleştirmek gerekir. Aşağıdaki bölüm OpenSUSE, Fedora ve Ubuntu için OpenID kimlik doğrulamasını etkinleştirir:

```
# 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",
)
```

Ayrıca bakınız:

[OpenID](#)

GitHub kimlik doğrulaması

GitHub üzerinde bir OAuth uygulaması oluşturduktan sonra tüm erişim bilgilerini Weblate üzerine yazmanız gerekir:

```
# 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, geri çağırma adresi “<https://ornek.com/accounts/complete/github/>” olacak biçimde yapılandırılmalıdır.

Kuruluşlar için GitHub ve Takımlar için GitHub benzer kimlik doğrulama arka yüzleri vardır. Ayarları `SOCIAL_AUTH_GITHUB_ORG_*` ve `SOCIAL_AUTH_GITHUB_TEAM_*` olarak adlandırılmıştır ve `SOCIAL_AUTH_GITHUB_ORG_NAME` ya da `SOCIAL_AUTH_GITHUB_TEAM_ID` kapsamları için ek ayarlara gerek duyarlar. Geri çağırma adresleri `https://example.com/accounts/complete/github-org/` ve `https://example.com/accounts/complete/github-teams/` biçimindedir.

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu uyarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

[GitHub](#)

Bitbucket kimlik doğrulaması

Bitbucket üzerinde bir OAuth uygulaması oluşturduktan sonra tüm erişim bilgilerini Weblate üzerine yazmanız gerekir:

```
# 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
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

Bitbucket

Google OAuth 2

Google OAuth 2 kullanmak için <<https://console.developers.google.com/>> adresinden bir uygulama kaydedip Google+ API uygulamasını etkinleştirmeniz gerekir.

Yönlendirme adresi: `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"
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

Google

Facebook OAuth 2

OAuth 2 hizmetlerinde her zamanki gibi, uygulamanızı Facebook üzerine kaydetmeniz gerekir. Bunu yaptıktan sonra, Weblate yapılandırmasını uygulamayı kullanacak biçimde ayarlayabilirsiniz:

Yönlendirme adresi: `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"]
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: *[Doğru site etki alanını ayarlayın](#)*.

Ayrıca bakınız:

Facebook

GitLab OAuth 2

GitLab OAuth 2 kullanmak için `<https://gitlab.com/profile/applications>` adresinden bir uygulama kaydetmeniz gerekir.

Yönlendirme adresi: `https://WEBLATE_SERVER/accounts/complete/gitlab/` and ensure you mark the `read_user` scope.

```
# 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/'
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: *[Doğru site etki alanını ayarlayın](#)*.

Ayrıca bakınız:

GitLab

Microsoft Azure Active Directory

Weblate, kimlik doğrulaması için genel ya da belirli kiracıları kullanacak şekilde yapılandırılabilir.

Yönlendirme adresi: Genel kiracılar için `https://WEBLATE SERVER/accounts/complete/azuread-oauth2/` ve belirli kiracılar için `https://WEBLATE SERVER/accounts/complete/azuread-tenant-oauth2/` biçimindedir.

```
# 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 = ""
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

Microsoft Azure Active Directory

Slack

Slack OAuth 2 kullanmak için `<https://api.slack.com/apps>` adresinden bir uygulama kaydetmeniz gerekir.

Yönlendirme adresi: `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 = ""
```

Not: Weblate, kimlik doğrulaması yapılırken belirtilen geri çağırma adresinde yapılandırılmış etki alanı bulunur. Adres uyumsuzluğuyla ilgili sorunlar görürseniz bu ayarı düzeltmek isteyebilirsiniz. Bilgi almak için: [Doğru site etki alanını ayarlayın](#).

Ayrıca bakınız:

[Slack](#)

Kimlik doğrulama yöntemi adlarını ve simgelerini değiştirmek

Kimlik doğrulama yönteminin görüntülenen adını ve simgesini `SOCIAL_AUTH_<NAME>_IMAGE` ve `SOCIAL_AUTH_<NAME>_TITLE` ayarları ile değiştirebilirsiniz. Örneğin Auth0 adı değiştirildiğinde şöyle görüntülenir:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Parola ile kimlik doğrulamayı kapatmak

E-posta ve parola kimlik doğrulaması `AUTHENTICATION_BACKENDS` bölümündeki `social_core.backends.email.EmailAuth` seçeneği kaldırılarak kapatılabilir. Temel Weblate işlevleri için gerekli olduğundan `weblate.accounts.auth.WeblateUserBackend` seçeneğini her zaman orada tutun.

E-posta kimlik doğrulamasını devre dışı bırakmak, e-postay ile ilgili tüm işlevleri (kullanıcı daveti ya da parola sıfırlama özelliği) devre dışı bırakır.

Tüyo: Yönetici arabirimi için, orada el ile oluşturduğunuz kullanıcılarda parola ile kimlik doğrulamasını kullanmayı sürdürebilirsiniz. ```/admin/login/``` adresine gitmeniz yeterlidir.

Örneğin, yalnızca openSUSE Open ID hizmet sağlayıcısını kullanan kimlik doğrulaması şunu kullanarak yapılabilir:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Parola ile kimlik doğrulama

Varsayılan `settings.py` dosyasında mantıklı bir `AUTH_PASSWORD_VALIDATORS` kümesi bulunur:

- Parola, kişisel bilgilerinize çok benzer bir şey olamaz.
- Parola en az 10 karakter uzunluğunda olmalıdır.
- Parola yaygın kullanılan bir şey olamaz.
- Parola tümüyle rakamlardan oluşamaz.
- Parola tek bir karakterden ya da boşluk karakterinden oluşamaz.
- Parola daha önce kullandığınız bir parola olamaz.

Bu seçeneği parola ilkenize uygun biçimde değiştirebilirsiniz.

Ek olarak, parola zorluğu ile ilgili oldukça gerçekçi öngörüler veren ve belirli bir eşğin altında kalan parolaların reddedilmesini sağlayan `django-zxcvbn-password` dosyasını da yükleyebilirsiniz.

2.5.5 SAML kimlik doğrulaması

4.1.1 sürümünde geldi.

Lütfen yapılandırma için Python Social Auth yönergelerini izleyin. Önemli farklar:

- Weblate, ``SOCIAL_AUTH_SAML_ENABLED_IDPS`` içinde `weblate` olarak adlandırılması gereken tek kimlik hizmeti sağlayıcısını destekler.
- SAML XML üst verileri adresi: `/accounts/metadata/saml/`.
- Şu seçenekler kendiliğinden doldurulur: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Örnek yapılandırma:

```
# 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"
    }
}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}
```

Varsayılan yapılandırma, şu özniteliklerden kullanıcı bilgilerini ayıklar, kimlik hizmeti sağlayıcınızı bu öznitelikleri sunacak şekilde yapılandırın:

Öznitelik	SAML URI referansı
Ad soyad	urn:oid:2.5.4.3
Ad	urn:oid:2.5.4.42
Soyad	urn:oid:2.5.4.4
E-posta	urn:oid:0.9.2342.19200300.100.1.3
Kullanıcı adı	urn:oid:0.9.2342.19200300.100.1.1

İpucu: Yukarıdaki örnek ve Docker görüntüsü `weblate` adlı bir kimlik hizmeti sağlayıcısı tanımlamaktadır. Bu dizgeyi kimlik hizmeti sağlayıcınızda *Relay* olarak yapılandırmanız gerekebilir.

Ayrıca bakınız:

Docker içinde SAML yapılandırması, SAML

2.5.6 LDAP kimlik doğrulaması

LDAP kimlik doğrulaması en iyi şekilde *django-auth-ldap* paketi kullanılarak sağlanabilir. Her zamanki gibi kurabilirsiniz:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

İpucu: Bu paket Docker kapsayıcısı ile birlikte gelir. Bilgi almak için: *Docker ile kurmak*.

Not: Python LDAP 3.1.0 modülünde, bu sürümü kullanmanızı engelleyebilecek bazı uyumsuzluklar vardır. `AttributeError: 'module' nesnesinin '_trace_level' özniteliği yok` hatasıyla karşılaşsanız, `python-ldap` modülünü 3.0.0 sürümüne düşürmek yardımcı olabilir.

Paketi kurduktan sonra, Django kimlik doğrulaması ile bağlayabilirsiniz:

```
# 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",
)

# 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
```


Not: `:setting:django:AUTHENTICATION_BACKENDS` ayarından `''social_core.backends.email.EmailAuth` seçeneğini kaldırmalısınız. Yoksa kullanıcılar Weblate üzerinde kendi parolalarını ayarlayabilir ve bunu kullanarak kimlik doğrulaması yapabilir. İzinler almak ve anonim kullanıcılara kolaylık sağlamak için `weblate.accounts.auth.WeblateUserBackend` seçeneğinin tutulması hala gereklidir. Ayrıca, yerel bir yönetici hesabı oluşturduysanız (örneğin `:d jadmin:createadmin` kullanarak) oturum açmanızı da sağlar.

BIND parolasını kullanarak

Kimlik doğrulaması için doğrudan BIND kullanamıyorsanız, aramayı kullanmanız ve arama için bağlanacak bir kullanıcı belirtmeniz gerekir. Örneğin:

```
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)"
)
```

Active Directory bütünleştirilmesi

```
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
```

Ayrıca bakınız:

Django Authentication Using LDAP, Authentication

2.5.7 CAS kimlik doğrulaması

CAS kimlik doğrulaması, *django-cas-ng* gibi bir paket kullanılarak yapılabilir.

Birinci adım, kullanıcının e-posta alanını CAS aracılığıyla açıklamaktır. Bunun CAS sunucusunun kendisinde yapılandırılması gerekir ve CAS v1 öznelikleri tarafından desteklenmediğinden en az CAS v2 çalıştırmanız gerekir.

İkinci adım, Weblate yapılandırmasını CAS sunucunuzu ve özneliklerinizi kullanacak şekilde güncellemektir.

django-cas-ng kurmak için:

```
pip install django-cas-ng
```

Paketi kurduktan sonra, `settings.py` dosyasını değiştirerek Django kimlik doğrulama sistemiyle bağlantı kurabilirsiniz:

```
# 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")
```

Son olarak, e-posta alanını kullanıcı nesnesiyle eşlemek için bir sinyal kullanabilirsiniz. Bunun çalışması için sinyali *django-cas-ng* paketinden içe aktarmanız ve kodunuzu bu sinyale bağlamanız gerekir. Bunu ayarlar dosyasında yapmak sorunlara neden olabilir. Bu nedenle şunları koymanız önerilir:

- Uygulamanızın yapılandırmasında `:p y:meth: django:django.apps.AppConfig.ready` yöntemi
- Projenin `urls.py` dosyasında (herhangi bir model var olmadığında)

```
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()
```

Ayrıca bakınız:

Django CAS NG

2.5.8 Üçüncü taraf Django kimlik doğrulamasını yapılandırmak

Genellikle herhangi bir Django kimlik doğrulama uygulama eki Weblate ile çalışmalıdır. Yalnızca eklentinin yönergelerini izleyin. Weblate kullanıcı arka ucunu kurulmuş olarak tutmayı unutmayın.

Ayrıca bakınız:

LDAP kimlik doğrulaması, CAS kimlik doğrulaması

Genellikle kurulum, `AUTHENTICATION_BACKENDS` seçeneğine bir kimlik doğrulama arka ucu eklemekten ve `INSTALLED_APPS` içine bir kimlik doğrulama uygulaması (varsa) kurmaktan oluşur:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Erişim denetimi

Weblate, kullanıcı izinlerinin uygulamanın tamamı ya da bir bölümü için atanabileceği ayrıntılı bir yetki sistemi sunar.

3.0 sürümünde değişti: Weblate 3.0 sürümünden önce, yetki sistemi yalnızca Django izin sistemine dayanıyordu. Ancak şimdi Weblate uygulamasına özel olarak oluşturuldu. Daha eski bir sürüm kullanıyorsanız, lütfen kullandığınız sürümün belgelerine bakın.

2.6.1 Basit erişim denetimi

Tüm Weblate kurulumunu yönetmiyorsanız ve yalnızca belirli projelere erişiminiz varsa ([Hosted Weblate](#) gibi), erişim denetimi yönetimi ayarlarınız aşağıdaki seçeneklerle kısıtlıdır. Daha karmaşık bir kurulumla gerek duymuyorsanız, bunlar sizin için yeterlidir.

Proje erişim denetimi

Not: Bu özellik, Hosted Weblate üzerinde Libre tarifesini kullanan projeler için kullanılamaz.

Farklı bir *Erişim denetimi* ayarı seçerek kullanıcının bireysel projelere erişimini sınırlayabilirsiniz. Kullanılabilecek seçenekler şunlardır:

Herkese açık

Herkese görebilir, tüm oturum açmış kullanıcılar çeviri yapabilir.

Korunmuş

Herkese görebilir, yalnızca seçilmiş kullanıcılar çeviri yapabilir.

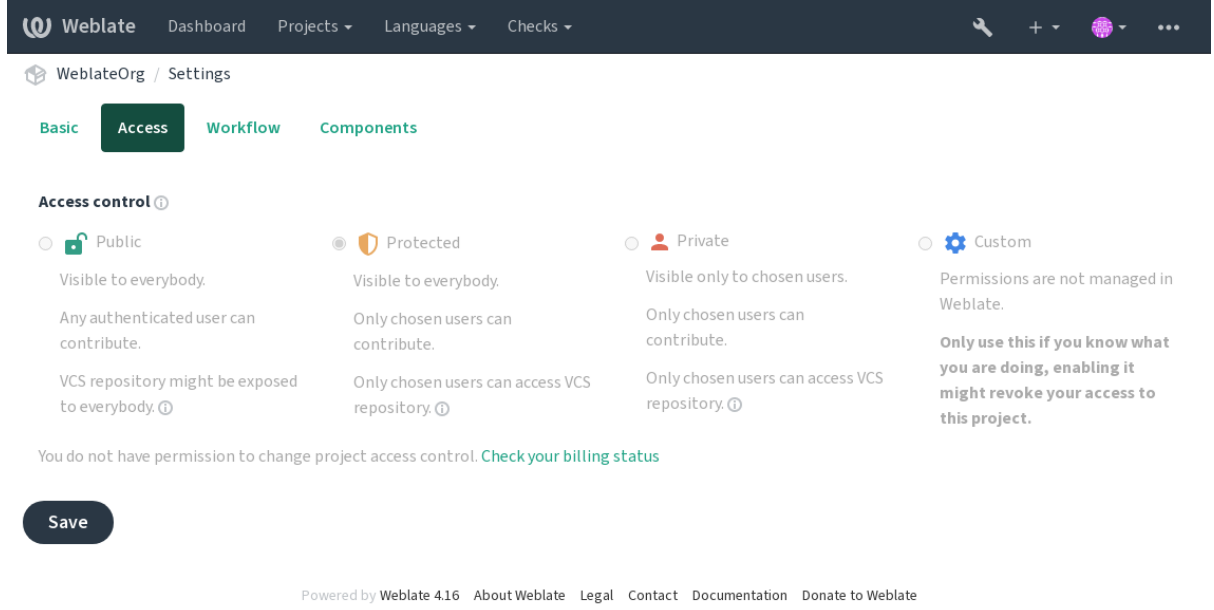
Kişisel

Yalnızca seçilmiş kullanıcılar görebilir ve çeviri yapabilir.

Özel

Kullanıcı yönetimi özellikleri devre dışı bırakılır. Varsayılan olarak tüm kullanıcıların proje üzerinde herhangi bir işlem yapması engelenir. Tüm izinlerin *Özel erişim denetimi* kullanılarak ayarlanması gerekir.

Bir projenin *Erişim denetimi* yapılandırması, ilgili projenin (*Yönetim* ↓ *Ayarlar*) *Erişim* sekmesinde değiştirilebilir.



Varsayılan değer `DEFAULT_ACCESS_CONTROL` ile değiştirilebilir.

Not: *Kişisel* projeler için bile, projenizle ilgili bazı bilgiler açığa çıkar: Uygulama kopyasının tamamı için istatistikler ve dil özetinde, erişim denetimi ayarına rağmen tüm projelerin sayıları bulunur. Proje adınız ve diğer bilgileriniz açığa çıkmaz.

Not: *Herkese açık*, *Korunmuş* ve *Kişisel* projelerde kullanıcılar için varsayılan olarak var olan gerçek izinler kümesi, Weblate kopyasının yöneticisi tarafından *özel ayar* kullanılarak yeniden tanımlanabilir.

Ayrıca bakınız:

[Erişim denetimi](#)

Her proje için erişim denetimi yönetimi

Proje erişimi yönetimi yetkisi olan kullanıcılar (bilgi almak için [Yetkiler ve hazır rollerin listesi](#)), projelerdeki kullanıcıları takımlara ekleyerek yönetebilir. İlk takım grubu Weblate tarafından sağlanır. Daha ayrıntılı erişim denetimi sağlamak için ek takımlar tanımlanabilir. Ekipleri dillerle sınırlayabilir ve onlara belirli erişim rolleri atayabilirsiniz (ayrıntılı bilgi almak için [Yetkiler ve hazır rollerin listesi](#)).

Her proje için şu takımlar kendiliğinden oluşturulur:

Herkese açık, *Korunmuş* ve *Kişisel* projeler için:

Yönetim

Proje için kullanılacak tüm izinleri kapsar.

Onaylama (yalnızca *onaylama iş akışı* özelliği açıksa)

Çevirileri gözden geçirip onaylayabilir.

Yalnızca *Korunmuş* ve *Kişisel* projeler için:

Çevir

Projeyi çevirebilir ve çevrim dışı yapılmış çevirileri yükleyebilir.

Kaynaklar

(*Proje ayarları* içinde izin verilmiş ise) kaynak dizgeleri ve kaynak dizge bilgilerini düzenleyebilir.

Diller

Çevrilecek dilleri yönetebilir (çevirileri ekleyebilir ya da kaldırabilir).

Sözlük

Sözlükleri yönetebilir (kayıtları ekleyebilir ya da kaldırabilir ve yükleyebilir).

Bellek

Çeviri belleğini yönetebilir.

Ekran görüntüleri

Ekran görüntülerini yönetebilir (ekleyip kaldırabilir ve kaynak dizgelerle ilişkilendirebilir).

Kendiliğinden çeviri

Kendiliğinden çevirileri kullanabilir.

Sürüm denetimi sistemi

Sürüm denetimi sisteminin yönetimi yapabilir ve dışa aktarılmış depoya erişebilir.

Faturalama

Faturalama bilgilerine ve ayarlarına erişebilir (bilgi almak için [Faturalama](#)).

Bu özellikler, projenin *Yönetim* ↓ *Kullanıcılar* menüsünden erişilebilen *Erişim denetimi* sayfasında bulunur.

Takım yöneticileri

4.15 sürümünde geldi.

Her takımda, takıma kullanıcı ekleyip kaldırabilen bir takım yöneticisi olabilir. Bu seçenek, kendi kendini yöneten takımlar oluşturmak istediğinizde kullanışlıdır.

Yeni kullanıcı daveti

Var olan bir kullanıcıyı projeye eklemenin yanında yeni kullanıcılar davet edilebilir. Yeni kullanıcılar hemen oluşturulur. E-posta yoluyla gönderilen davetlerde ise kullanıcı davet içindeki bağlantıyı kullanarak oturum açana kadar hesap etkinleştirilmez. Bunu yapabilmek için site genelinde herhangi bir yetki verilmesi gerekmez. Proje kapsamında erişim yönetimi izni (*Yönetim* takımının üyesi olmak gibi) yeterli olur.

İpucu: Davet edilen kullanıcı davetiyenin geçerlilik süresini geçirdiyse, hesap zaten oluşturulmuş olduğundan, parola sıfırlama formunda davet edildiği e-posta adresini kullanarak parolasını ayarlayabilir.

3.11 sürümünde geldi: Kullanıcı davet e-postaları yeniden gönderilebilir (önceki davet e-postası geçersiz olur).

Aynı türde davetler site genelinde *Kullanıcılar* sekmesindeki *yönetim bölümü* içinden kullanılabilir.

Kullanıcıları engelleme

4.7 sürümünde geldi.

Projenizdeki bazı kullanıcıların kötü davranışlar sergilemesi durumunda, katkıda bulunmalarını engelleme seçeneğiniz vardır. Engellenen kullanıcının izinleri varsa, projeyi görmeye sürdürebilir. Ancak katkıda bulunamaz.

Her projeye özel izin yönetimi

Projelerinizi *Korunmuş* ya da *Kişisel* olarak ayarlayabilir ve Weblate kullanıcı arayüzünde her proje için *kullanıcıları yönetebilirsiniz*.

Varsayılan olarak bu durum, bu takımların kendi yapılandırması nedeniyle Weblate tarafından *Kullanıcılar* ve *Görüntüleyenler* :ref: varsayılan takımlar <default-teams> için sunulan erişim izinlerinin verilmesini engeller. Bu durum, varsayılan takımları değiştirerek, yeni bir tane oluşturarak ya da aşağıdaki *Özel erişim denetimi* bölümünde açıklandığı gibi tek tek bileşenler için ek özel ayarlar oluşturarak site genelinde bu projelere izin vermenizi engellemez.

Weblate kullanıcı arayüzü üzerinden izinleri yönetmenin en önemli avantajlarından biri, izinlerin diğer kullanıcılara süper kullanıcı yetkisi verilmeden devredilebilmesidir. Bunun için bu kullanıcıları projenin *Yönetim* ekibine ekleyin.

2.6.2 Özel erişim denetimi

Not: Bu özellik, Hosted Weblate üzerinde Libre tarifesini kullanan projeler için kullanılamaz.

İzin sistemi takımlara ve rollere dayanır. Roller bir dizi izin tanımlar ve takımlar bunları kullanıcılara ve çevrilere bağlar. Bilgi almak için *Kullanıcılar*, *roller*, *takımlar* ve *izinler* bölümüne bakabilirsiniz.

Weblate erişim denetimi sisteminin şimdilik en güçlü özellikleri yalnızca *Django yönetim arayüzü* üzerinden kullanılabilir. Bu arayüzden herhangi bir projenin izinlerini yönetebilirsiniz. Kullanabilmek için *Erişim denetimini* mutlaka *Özel* olarak değiştirmeniz gerekmez. Ancak süper kullanıcı yetkilerinizin olması gerekir.

Uygulamanın ayrıntılarıyla ilgilenmiyorsanız ve yalnızca varsayılan ayarlara dayanan yeterince basit bir yapılandırma oluşturmak istiyorsanız ya da Weblate kurulumunun tamamına site genelinde erişiminiz yoksa (*Hosted Weblate* gibi), lütfen *Basit erişim denetimi* bölümüne bakın.

Sık kullanılan kurulumlar

Bu bölümde, ilginizi çekebilecek bazı genel yapılandırmaların özetini bulabilirsiniz.

Site genelinde izin yönetimi

Uygulama kopyasının tamamındaki izinleri bir kerede yönetmek için kullanıcıları uygun *Varsayılan takımlar* ögesine ekleyin:

- *Kullanıcılar* (bu varsayılan olarak *kendiliğinden takım ataması* ile yapılır).
- *Onaylayanlar* (:ref: *onaylayan kişiler ile onaylama iş akışını* <reviews> kullanıyorsanız).
- *Yöneticiler* (yönetim işlemlerinin çoğunu başkasına devretmek istiyorsanız).

Tüm projeleri *Herkese açık* olarak yapılandırılmış tutmalısınız (bilgi almak için *Proje erişim denetimi*). Yoksa *Kullanıcılar* ve *Onaylayanlar* takımlarına üye olarak elde edilen site geneli izinlerin herhangi bir etkisi olmaz.

Varsayılan takımlara seçtiğiniz bazı ek izinleri de verebilirsiniz. Örneğin, tüm *Kullanıcılara* ekran görüntülerini yönetme izni vermek isteyebilirsiniz.

Bazı yeni özel takımlar da tanımlayabilirsiniz. Bu takımlar için izinlerinizi site genelinde yönetmeyi sürdürmek istiyorsanız, *Proje seçimi* için uygun bir değer seçin (*Tüm projeler* ya da *Tüm herkese açık projeler* gibi).

Diller, bileşenler veya projeler için özel izinler

Diller, bileşenler ve projeler gibi farklı nesnelerin izinlerini yönetmek için kendi ayrılmış takımlarınızı oluşturabilirsiniz. Bu takımlar yalnızca ek izinler verebilse de, başka bir özel takım ekleyerek site genelinde ya da her proje için takımlar tarafından verilen izinleri iptal edemezsiniz.

Örnek:

(Nedeni ne olursa olsun) belirli bir dilde çevirinin (*Çekçe* gibi) yalnızca belirli güvenilir bir çevirmen grubu tarafından yapılmasına izin vermek ve diğer dillerde yapılan çevirileri herkese açık tutmak istiyorsanız şunu yapmanız gerekir:

1. Tüm kullanıcılardan *Çekçe* dilini çevirme iznini kaldırın. Varsayılan yapılandırmada, *Kullanıcılar* *varsayılan takım* değiştirilerek yapılabilir.

Tablo 1: *Kullanıcılar* grubu

Dil seçimi	Tanımlandığı gibi
Diller	<i>Çekçe</i> dışında tümü

2. *Çekçe* çevirmenleri için özel bir takım ekleyin.

Tablo 2: *Çekçe çevirmenleri* grubu

Roller	<i>Uzman kullanıcılar</i>
Proje seçimi	<i>Tüm herkese açık projeler</i>
Dil seçimi	<i>Tanımlandığı gibi</i>
Diller	<i>Çekçe</i>

3. İzin vermek istediğiniz kullanıcıları bu takıma ekleyin.

Gördüğünüz gibi, bu şekilde izin yönetimi güçlüdür. Ancak oldukça sıkıcı bir iş olabilir. Süper kullanıcı izinleri vermediğiniz sürece izinleri başka bir kullanıcıya devredemezsiniz.

Kullanıcılar, roller, takımlar ve izinler

Kimlik doğrulama modelleri birkaç öğeden oluşur:

İzin

Weblate tarafından tanımlanan bireysel izin. İzinler kullanıcılara atanamaz. Bu yalnızca rol ataması ile yapılabilir.

Rol

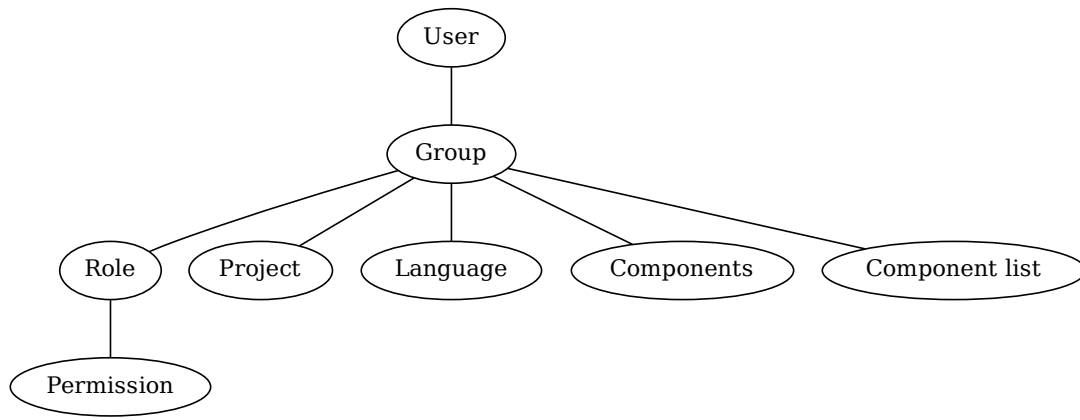
Rol, bir izin kümesi tanımlar. Böylece, bu kümeler birkaç yerde yeniden kullanılabilir ve yönetim kolaylaşır.

Kullanıcı

Kullanıcı birkaç takımın üyesi olabilir.

Grup

Gruplar, roller, kullanıcılar ve kimlik doğrulama nesneleri (projeler, diller ve bileşen listeleri) arasında bağlantı kurar.



Not: Bir takımın kendisine atanmış herhangi bir rolü olmayabilir. Bu durumda projeye herhangi birinin göz atma erişimi olacağı varsayılır (aşağıya bakın).

Bir projeye göz atma erişimi

Kullanıcı, projeye bağlı bir takımın veya bu projenin içindeki herhangi bir bileşenin üyesi olmalıdır. Üye olmak yeterlidir. Projeye göz atabilmek için ayrıca özel bir izin gerekmez (bu, varsayılan *Görüntüleyiciler* takımında kullanılır. Bilgi almak için [Takım listesi](#)).

Bir bileşene göz atma erişimi

Bir kullanıcı, bileşenlerin projesine erişebildiğinde kısıtlanmamış bileşenlere erişebilir (ve kullanıcıya proje için verilen tüm izinlere sahip olur). *Kısıtlı erişim* açıkken, bileşene erişim, bileşen (veya bileşenin içinde bulunduğu bileşen listesi) için açık izinler gerekir.

Takımların kapsamı

Takımlardaki roller tarafından atanan izin kapsamı şu kurallara göre uygulanır:

- Takım herhangi bir *Bileşen listesi* belirtirse, takıma bağlı bileşen listelerindeki tüm bileşenler için o takımın üyelerine tüm izinler verilir ve bu bileşenlerin içinde bulunduğu tüm projeler için ek izinleri olmayan bir erişim verilir. *Bileşenler* ve *Projeler* yok sayılır.
- Takımda herhangi bir *Bileşen* belirtiliyorsa, takıma bağlı tüm bileşenler için o takımın üyelerine tüm izinler verilir ve bu bileşenlerin içinde bulunduğu tüm projeler için ek izinleri olmayan bir erişim verilir. *Projeler* yok sayılır.
- Yoksa, takımda herhangi bir *Proje* belirtiliyorsa, bunları doğrudan listeleyerek veya *Proje seçimi* değerini *Tüm herkese açık projeler* gibi bir değere ayarlayarak, tüm bu izinler tüm projelere uygulanır ve bu da tüm projelere erişmek için aynı izinleri etkin bir şekilde verir *kısıtlanmamış bileşenler*.
- Bir takımın *Diller* değeri tarafından uygulanan kısıtlamalar, kullanıcının belirli işlemleri yapma izni olup olmadığına bakılarak ayrı olarak uygulanır. Yani, yalnızca çeviri sürecinin kendisiyle doğrudan ilgili onaylama, çevirileri kaydetme, öneri ekleme gibi işlemlere uygulanır.

İpucu: Tüm dillerin veya projelerin kendiliğinden katılması için *Dil seçimi* ya da *Proje seçimi* kullanın.

Örnek:

Diyelim ki şu bileşenleri olan bir `foo` projesi var: `foo/bar` ve `foo/baz` ve şu takım:

Tablo 3: Grup *İspanyol Yönetici-Onaylayıcılar*

Roller	<i>Dizgeleri onayla, Depo yönetimi</i>
Bileşenler	<code>foo/bar</code>
Diller	<i>İspanyolca</i>

Bu takımın üyelerinin şu izinleri olacak (varsayılan rol ayarları kabul edilerek):

- Şu iki bileşenin de bulunduğu `foo` projesi için genel (göz atma) izni: `foo/bar` ve `foo/baz`.
- `foo/bar` *İspanyolca* çevirisinde (başka bir yerde değil) dizgeleri onaylama izni.
- Tüm `foo/bar` deposu için sürüm denetimi sistemi yönetimi. Çevirmenler tarafından tüm diller için yapılmış bekleyen değişikliklerin işlenmesi gibi.

Kendiliğinden takım atamaları

Django yönetim arayüzü içindeki *Grup* düzenleme sayfasının alt bölümünde, yeni oluşturulan kullanıcıları e-posta adreslerine göre bir takıma kendiliğinden atanması için kullanılacak kurallı ifade listesi ile *Kendiliğinden takım ataması* yapabilirsiniz. Bu atama yalnızca hesap oluşturulduktan sonra yapılır.

Bu özelliğin en yaygın kullanım örneği, tüm yeni kullanıcıların bazı varsayılan takımlara atanmasıdır. Bunu yapmak için, büyük olasılıkla kurallı ifade alanındaki varsayılan değeri (`^^.*$`) korumak isteyeceksiniz. Bu seçeneğin başka bir kullanım örneği, varsayılan olarak şirketinizin çalışanlarına bazı ek ayrıcalıklar vermek olabilir. Hepsinin etki alanınızdaki kurumsal e-posta adreslerini kullandığını varsayarsak, bu `^.*@mycompany.com` gibi bir ifadeyle gerçekleştirilebilir.

Not: *Kullanıcılar* ve *Görüntüleyiciler* için kendiliğinden takım ataması, bir Weblate sürümünden diğerine yükseltme yaparken her zaman yeniden oluşturulur. Bu seçeneği kapatmak istiyorsanız, kurallı ifadeyi `^$` olarak ayarlayın (hiçbir şeyle eşleşmez).

Not: Şimdilik, zaten var olan kullanıcıları kullanıcı arayüzü üzerinden bazı takımlara toplu olarak eklemenin bir yolu yok. Bunun için [REST API](#) kullanmayı deneyebilirsiniz.

Varsayılan takımlar ve roller

Kurulundan sonra, bir varsayılan takım kümesi oluşturulur (bilgi almak için [Takım listesi](#)).

Bu roller ve takımlar kurulum sonrasında oluşturulur. Hazır roller, yükseltme yapılırken veri tabanı aktarımı tarafından her zaman güncel tutulur. Bunları gerçekten değiştiremezsiniz. Kendi izin kümenizi tanımlamak istiyorsanız lütfen yeni bir rol tanımlayın.

Yetkiler ve hazır rollerin listesi

Kapsam	İzin	Roller
Faturalama (bilgi almak için Faturalama)	Fatura bilgilerini görüntüle	<i>Yönetim, Faturalama</i>
Değişiklikler	Değişiklikleri indir	<i>Yönetim</i>
Yorumlar	Yorum gönderme	<i>Yönetim, Kaynağı düzenleme, Uzman kullanıcı</i>
	Yorum silme	<i>Yönetim</i>
	Yorumu çözümleme	<i>Yönetim, Dizgeleri onaylama</i>
Bileşen	Bileşen ayarlarını düzenleme	<i>Yönetim</i>
	Bileşeni kilitleme, çevirileri engelleme	<i>Yönetim</i>
Sözlük	Sözlük kaydı ekleme	<i>Yönetim, Sözlük yönetimi, Uzman kullanıcı</i>
	Sözlük kaydını düzenleme	<i>Yönetim, Sözlük yönetimi, Uzman kullanıcı</i>
	Sözlük kaydını silme	<i>Yönetim, Sözlük yönetimi, Uzman kullanıcı</i>
	Sözlük kayıtları yükleme	<i>Yönetim, Sözlük yönetimi, Uzman kullanıcı</i>
Kendiliğinden öneriler	Kendiliğinden önerileri kullanma	<i>Yönetim, Kaynağı düzenleme, Uzman kullanıcı</i>
Çeviri belleği	Çeviri belleğini düzenleme	<i>Yönetim, Çeviri belleği yönetimi</i>
	Çeviri belleğini silme	<i>Yönetim, Çeviri belleği yönetimi</i>
Projeler	Proje ayarlarını düzenleme	<i>Yönetim</i>
	Proje erişimini yönetme	<i>Yönetim</i>
Raporlar	Raporları indirme	<i>Yönetim</i>
Ekran görüntüleri	Ekran görüntüsü ekleme	<i>Yönetim, Ekran görüntüleri yönetimi</i>
	Ekran görüntüsü düzenleme	<i>Yönetim, Ekran görüntüleri yönetimi</i>
	Ekran görüntüsü silme	<i>Yönetim, Ekran görüntüleri yönetimi</i>
Kaynak dizgeleri	Ek dizge bilgilerini düzenleme	<i>Yönetim, Kaynağı düzenleme</i>
Dizgeler	Yeni dizge ekleme	<i>Yönetim</i>
	Bir dizgeyi silme	<i>Yönetim</i>
	Denetim hatasını yok sayma	<i>Yönetim, Kaynağı düzenleme, Uzman kullanıcı</i>
	Dizgeleri düzenleme	<i>Yönetim, Kaynağı düzenleme, Uzman kullanıcı</i>
	Dizgeleri onaylama	<i>Yönetim, Dizgeleri onaylama</i>
	Öneriler dayatıldığında dizgeyi düzenleme	<i>Yönetim, Dizgeleri onaylama</i>
	Kaynak dizgeleri düzenleme	<i>Yönetim, Kaynağı düzenleme, Uzman kullanıcı</i>
Öneriler	Öneriyi kabul etme	<i>Yönetim, Kaynağı düzenleme, Uzman kullanıcı</i>
	Öneri ekleme	<i>Yönetim, Kaynağı düzenleme, Öneri ekleme</i>
	Öneri silme	<i>Yönetim, Uzman kullanıcı</i>
	Öneriye oy verme	<i>Yönetim, Kaynağı düzenleme, Uzman kullanıcı</i>
Çeviriler	Çeviri dili ekleme	<i>Yönetim, Uzman kullanıcı, Dil yönetimi</i>
	Kendiliğinden çevirme	<i>Yönetim, Kendiliğinden çevirme</i>
	Var olan çeviriyi silme	<i>Yönetim, Dil yönetimi</i>
	Çeviri dosyasını indirme	<i>Yönetim, Kaynağı düzenleme, Depoya yükleme</i>
	Çevirilecek birkaç dil ekleyin	<i>Yönetim, Dil yönetimi</i>
Yüklemeler	Yüklenmiş çeviriye katkıda bulunanı tanımlama	<i>Yönetim</i>

Tablo 4 – önceki sayfadan devam

Kapsam	İzin	Roller
Sürüm denetimi sistemi	Yükleme ile var olan dizgelerin üzerine yazma	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
	Çeviri yükleme	<i>Yönetim, Kaynağı düzenleme, Uzman</i>
	İç depoya erişme	<i>Yönetim, Depoya erişme, Uzman kull</i>
	Değişiklikleri iç depoya işleme	<i>Yönetim, Depo yönetimi</i>
	İç depodaki değişiklikleri işleme	<i>Yönetim, Depo yönetimi</i>
	İç depodaki değişiklikleri sıfırlama	<i>Yönetim, Depo yönetimi</i>
	Yukarı akış deposu konumunu görüntüleme	<i>Yönetim, Depoya erişme, Uzman kull</i>
Site genelindeki izinler	İç depoyu güncelleme	<i>Yönetim, Depo yönetimi</i>
	Yönetim arayüzünü kullanma	
	Yeni projeler ekleme	
	Dil tanımları ekleme	
	Dil tanımları yönetimi	
	Takım yönetimi	
	Kullanıcı yönetimi	
	Rol yönetimi	
	Duyuru yönetimi	
	Çeviri belleği yönetimi	
	Makine yönetimi	
	Bileşen listeleri yönetimi	

Not: Site genelindeki izinler herhangi bir varsayılan role verilmez. Bu izinler güçlüdür ve süper kullanıcı yetkilerine oldukça yakındır. Çoğu Weblate kurulumundaki tüm projeleri etkiler.

Takım listesi

Aşağıdaki takımlar kurulumun ardından (ya da *setupgroups* çalıştırıldıktan sonra) oluşturulur ve bunları istediğiniz gibi değiştirebilirsiniz. Ancak, silerseniz ya da yeniden adlandırırsanız aktarım işlemi bunları yeniden oluşturur.

Konuklar

Kimliği doğrulanmamış kullanıcıların izinlerini belirler.

Bu takımda yalnızca anonim kullanıcılar bulunur (bilgi almak için *ANONYMOUS_USER_NAME*).

Kimliği doğrulanmamış kullanıcıların izinlerini sınırlamak için bu takımdaki rolleri kaldırabilirsiniz.

Varsayılan roller: *Öneri ekleme, Depoya erişme*

Görüntüleyiciler

Bu rol, herkese açık projelerin tüm kullanıcılara görüldüğünden emin olunmasını sağlar. Varsayılan olarak tüm kullanıcılar bu grubun üyesidir.

Varsayılan olarak, *kendiliğinden takım ataması*, katılan tüm yeni hesapları bu takımın bir üyesi yapar.

Varsayılan roller: Yok

Kullanıcılar

Tüm kullanıcıların varsayılan takımı.

Varsayılan olarak, *kendiliğinden takım ataması*, katılan tüm yeni hesapları bu takımın bir üyesi yapar.

Varsayılan roller: *Uzman kullanıcı*

Onaylayanlar

Onaylayan kullanıcıların takımı (bilgi almak için *Çeviri iş akışları*).

Varsayılan roller: *Dizgeleri onaylama*

Yöneticiler

Yönetici kullanıcıların takımı.

Varsayılan roller: *Yönetim*

Uyarı: Hazır gelen Weblate takımlarını ve kullanıcılarını asla silmeyin. Bu işlem beklenmedik sorunlara yol açabilir! Onları kullanmıyorsanız, silmek yerine tüm izinlerini kaldırabilirsiniz.

2.6.3 Ek erişim kısıtlamaları

Weblate kurulumunuzu herkese daha az açık olacak biçimde kullanmak, yani yeni kullanıcıları yalnızca davet ederek izin vermek istiyorsanız, Weblate yapılandırmasını yalnızca bilinen kullanıcıların erişebileceği şekilde ayarlayabilirsiniz. Bu amaçla, yeni kullanıcıların hesap açmasını önlemek için [REGISTRATION_OPEN](#) seçeneğini *Yanlış* olarak ve [REQUIRE_LOGIN](#) seçeneğini */'* olarak ayarlayarak tüm site sayfalarına erişmek için oturum açmayı zorunlu kılabilirsiniz. Böylece kolayca Weblate kurulumunuzu kilitlemiş olursunuz.

İpucu: Yeni kullanıcılar eklemek için iç *Yeni kullanıcı daveti* komutunu kullanabilirsiniz.

2.7 Çeviri projeleri

2.7.1 Çeviri kurulumu

Weblate organizes translatable VCS content of project/components into a tree-like structure.

- The bottom level object is *Proje yapılandırması*, which should hold all translations belonging together (for example translation of an application in several versions and/or accompanying documentation).
- On the level above, *Bileşen yapılandırması*, which is actually the component to translate, you define the VCS repository to use, and the mask of files to translate.
- Above *Bileşen yapılandırması* there are individual translations, handled automatically by Weblate as translation files (which match *Dosya maskesi* defined in *Bileşen yapılandırması*) appear in the VCS repository.

Weblate supports a wide range of translation formats (both bilingual and monolingual ones) supported by Translate Toolkit, see *Desteklenen dosya biçimleri*.

Not: You can share cloned VCS repositories using *İç Weblate adresleri*. 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

3.2 sürümünde değişti: An interface for adding projects and components is included, and you no longer have to use *Django yönetim arayüzü*.

3.4 sürümünde değişti: 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 *Faturalama*), 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:

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

[Your profile](#) / [Billing](#)

Billing plan	
Current plan	Basic plan (Active)
Monthly price	19 EUR
Yearly price	199 EUR
Strings limit	Used 0 <div></div>
Languages limit	Used 0 <div></div>
Last invoice	2023-02-28 - 2023-03-02
Projects limit	Used 0 of 1 <div></div>
Projects	No projects currently assigned! 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

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:

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

[Create project](#)

[Add new translation project](#)
[Import translation project](#)

Project name

Display name

URL slug

Name used in URLs and filenames.

Project website

Main website of translated project.

Translation instructions

https://weblate.org/contribute/

You can use Markdown and mention users by @username.

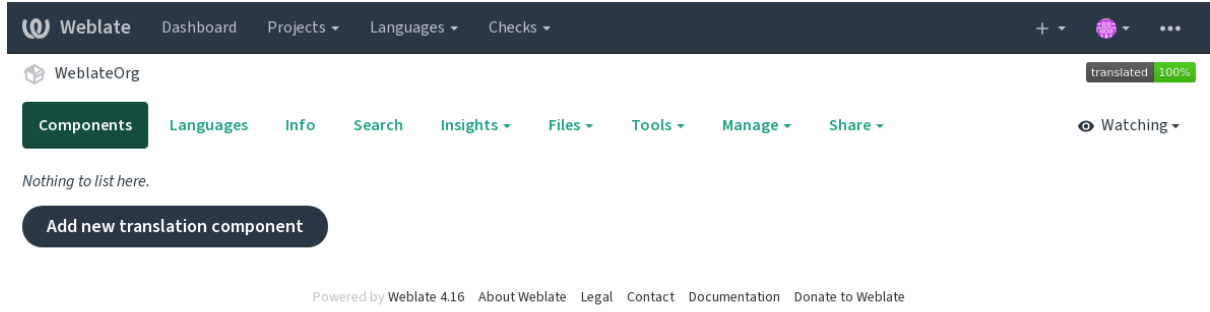
Billing

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:



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:

Sürüm denetiminden

Creates component from remote version control repository.

Var olan bileşenden

Creates additional component to existing one by choosing different files.

Ek dal

Creates additional component to existing one, just for different branch.

Çeviri dosyalarını yükle

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 [Weblate REST API uygulaması](#).

Belgeyi çevir

Upload single document or translation file and translate that.

Sıfırdan başla

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:

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

Create component

[From version control](#)
[Upload translations files](#)
[Translate document](#)
[Start from scratch](#)

Create a new translation component from remote version control system repository.

Component name ⓘ

Display name

URL slug ⓘ

Name used in URLs and filenames.

☐ Use as a glossary ⓘ

Project ⓘ

Source language ⓘ

Language used for source strings in all components

Version control system ⓘ

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 ⓘ

URL of a repository, use weblate://project/component to share it with other component.

Repository branch ⓘ

Repository branch to translate

Continue

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

On the next page, you are presented with a list of discovered translatable resources:

[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:

Create component

Detected license as MIT, please check whether it is correct.

Add new translation component

Project ⓘ
WeblateOrg

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/WeblateOrg/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/WeblateOrg/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

Ayrıca bakınız:*Django yönetim arayüzü, Proje yapılandırması, Bileşen yapılandırması*

2.7.3 Proje yapılandırması

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 *Çeviri belleği*.

Ayrıca bakınız:*/devel/integration*

These basic attributes set up and inform translators of a project:

Proje adı

Verbose project name, used to display the project name.

Ayrıca bakınız:*PROJECT_NAME_RESTRICT_RE***Adres kısaltması**

Project name suitable for URLs.

Proje web sitesi

URL where translators can find more info about the project.

This is a required parameter unless turned off by *WEBSITE_REQUIRED*.

Ayrıca bakınız:*PROJECT_WEB_RESTRICT_RE***Çeviri yönergeleri**

Text describing localization process in the project, and any other information useful for translators. Markdown can be used for text formatting or inserting links.

“Language-Team” üst bilgisi ayarlınsın

Whether Weblate should manage the Language-Team header (this is a *GNU gettext* only feature right now).

Paylaşılan çeviri belleği kullanılsın

Whether to use shared translation memory, see *Paylaşılan çeviri belleği* for more details.

Varsayılan değer `DEFAULT_ACCESS_CONTROL` seçeneği ile değiştirilebilir.

Paylaşılan çeviri belleğine katkıda bulunulsun

Whether to contribute to shared translation memory, see *Paylaşılan çeviri belleği* for more details.

Varsayılan değer `DEFAULT_ACCESS_CONTROL` seçeneği ile değiştirilebilir.

Erişim denetimi

Configure per project access control, see *Proje erişim denetimi* for more details.

Varsayılan değer `DEFAULT_ACCESS_CONTROL` ile değiştirilebilir.

Onaylama kullanılsın

Çeviriler için onaylama iş akışını etkinleştirir. Bilgi almak için: *Özel onaylayanlar*.

Kaynak onaylama kullanılsın

Kaynaklar için onaylama iş akışını etkinleştirir. Bilgi almak için: *Kaynak dizge onayları*.

Ayrıca bakınız:

report-source, *Yorumlar*

Kancalar kullanılsın

Whether unauthenticated *Bildirim kancaları* are to be used for this repository.

Ayrıca bakınız:

Ara dil dosyası, *Kaynak dizgeleri için kalite aracı*, *İki dilli ve tek dilli biçimler*, *Dil tanımları*

Dil kod adları

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`

İpucu: 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.

Ayrıca bakınız:

Yeni çevirileri eklemek, *Dil kodu*, *Dil kodlarını işlemek*

2.7.4 Bileşen yapılandırması

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.

Ayrıca bakınız:

/devel/integration

You can find some examples of typical configurations in the *Desteklenen dosya biçimleri*.

Not: 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:

Bileşen adı

Verbose component name, used to display the component name.

Bileşen kısaltması

Adresler için uygun bileşen adı.

Bileşen projesi

Proje yapılandırması bileşenin ait olduğu yer.

Sürüm denetimi sistemi

VCS to use, see *Sürüm denetimi bütünleştirmesi* for details.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri itmek

Kaynak kod deposu

VCS repository used to pull changes.

Ayrıca bakınız:

See *Depolara erişmek* for more details on specifying URLs.

İpucu: This can either be a real VCS URL or `weblate://project/component` indicating that the repository should be shared with another component. See *İç Weblate adresleri* for more details.

Depo itme adresi

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.

Ayrıca bakınız:

See *Depolara erişmek* for more details on how to specify a repository URL and *Weblate üzerindeki değişiklikleri itmek* for more details on pushing changes from Weblate.

Depo tarayıcısı

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use *Kalıp kodlaması*.

For example on GitHub, use something like: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

In case your paths are relative to different folder (path contains `..`), you might want to strip leading directory by `parentdir` filter (see *Kalıp kodlaması*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

Dışa aktarılmış depo adresi

URL where changes made by Weblate are exported. This is important when *Sürekli yerelleştirme* is not used, or when there is a need to manually merge changes. You can use *Git exporter* to automate this for Git repositories.

Depo dalı

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.

İtme işleminin yapılacağı dal

Branch for pushing changes, leave empty to use *Depo dalı*.

For linked repositories, this is not used and setting from linked component applies.

Not: This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

Ayrıca bakınız:

Weblate üzerindeki değişiklikleri itmek

Dosya maskesi

Mask of files to translate, including path. It should include one “*” replacing language code (see *Dil tanımları* 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.

For example `po/* .po` or `locale/*/LC_MESSAGES/django.po`.

In case your filename contains special characters such as `[,]`, these need to be escaped as `[\]` or `[\] \`.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, What does mean “There are more files for the single language (en)”?

Tek dilli temel dil dosyası

Base file containing string definitions for *Tek dilde bileşenler*.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, What does mean “There are more files for the single language (en)”?

Temel dosyayı düzenle

Whether to allow editing the base file for *Tek dilde bileşenler*.

Ara dil dosyası

Intermediate language file for *Tek dilde bileşenler*. 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 *Tek dilli temel dil dosyası*. In case the string is not translated into the source language, translating to other languages is prohibited. This provides *Kaynak dizgeler için kalite aracı*.

Ayrıca bakınız:

Kaynak dizgeler için kalite aracı, İki dilli ve tek dilli biçimler, What does mean “There are more files for the single language (en)”?

Yeni çeviriler kalıbı

Base file used to generate new translations, e.g. `.pot` file with gettext.

İpucu: 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.

Ayrıca bakınız:

adding-translation, *Yeni çevirileri eklemek, Yeni çeviri ekleme, İki dilli ve tek dilli biçimler, What does mean “There are more files for the single language (en)”?*

Dosya biçimi

Translation file format, see also *Desteklenen dosya biçimleri*.

Kaynak dizge hata bildirme adresi

Email address used for reporting upstream bugs. This address will also receive notification about any source string comments made in Weblate.

Çevirilerin yayılmasını sağlamak

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*.

Ayrıca bakınız:

Bileşenler arasında çevirilerin tutarlılığını sağlamak

Öneriler kullanılsın

Whether translation suggestions are accepted for this component.

Öneri oylaması

Turns on vote casting for suggestions, see *Öneri oylaması*.

Öneriler kendiliğinden kabul edilsin

Automatically accept voted suggestions, see *Öneri oylaması*.

Çeviri işaretleri

Customization of quality checks and other Weblate behavior, see *Parametreleri kullanarak davranışı özelleştirmek*.

Zorunlu denetimler

List of checks which can not be ignored, see *Denetimleri dayatmak*.

Not: Enforcing the check does not automatically enable it, you still should enabled it using *Parametreleri kullanarak davranışı özelleştirmek* in *Çeviri işaretleri* or *Additional info on source strings*.

Çeviri lisansı

License of the translation (does not need to be the same as the source code license).

Katılımcı anlaşması

Bir kullanıcının bu bileşeni çevirmeden önce kabul etmesi gereken katılımcı anlaşması.

Yeni çeviri ekleme

How to handle requests for creation of new languages. Available options:

Sorumlulara ulaşın

User can select desired language and the project maintainers will receive a notification about this. It is up to them to add (or not) the language to the repository.

Çeviri yönergeleri adresini göster

User is presented a link to page which describes process of starting new translations. Use this in case more formal process is desired (for example forming a team of people before starting actual translation).

Yeni dil dosyası oluştur

User can select language and Weblate automatically creates the file for it and translation can begin.

Yeni çevirilerin eklenmesi devre dışı bırakılmış

There will be no option for user to start new translation.

İpucu: The project admins can add new translations even if it is disabled here when it is possible (either *Yeni çeviriler kalıbı* or the file format supports starting from an empty file).

Ayrıca bakınız:

adding-translation, *Yeni çevirileri eklemek*

Dizge yönetimi

4.5 sürümünde geldi.

Configures whether users in Weblate will be allowed to add new strings and remove existing ones. Adjust this to match your localization workflow - how the new strings are supposed to be introduced.

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.

Ayrıca bakınız:

İki dilli ve tek dilli biçimler, adding-new-strings, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Dil kodu biçemi

Weblate tarafından oluşturulan çevirilerin dosya adını oluşturmakta kullanılacak dil kodunu özelleştirin.

Ayrıca bakınız:

Yeni çevirileri eklemek, Dil kodu, Dil kodlarını işlemek

Birleştirme biçemi

You can configure how updates from the upstream repository are handled. The actual implementation depends on VCS, see *Sürüm denetimi bütünleştirilmesi*.

Yeniden yerleştir

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 (itme dayatması ile)* as *Sürüm denetimi sistemi*, especially when pushing to a different branch.

Birleştir

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.

Hızlı ilerlemeden birleştir

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`.

İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri

Bir çeviri işlenirken kullanılacak ileti. Bilgi almak için: `:ref:<markup>`.

Varsayılan değerler şu seçeneklerle değiştirilebilir `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE`, `DEFAULT_MERGE_MESSAGE`, `DEFAULT_PULL_MESSAGE`.

İşleme ile itme

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 commit işlemleri*). To actually enable pushing *Repository push URL* has to be configured as well.

İşlenecek değişikliklerin yaşı

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.

Varsayılan değer `COMMIT_PENDING_HOURS` seçeneği ile değiştirilebilir.

İpucu: Bekleyen değişikliklerin işlenebileceği başka durumlar da vardır. Bilgi almak için: *Lazy commit işlemleri*.

Hata durumunda kilitlenme

Locks the component (and linked components, see [İç Weblate adresleri](#)) 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.

Herhangi bir depo hatası kalmadığında bileşenin kilidi kendiliğinden açılır.

Kaynak dil

Language used for source strings. Change this if you are translating from something else than English.

İpucu: 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 [Ara dil dosyası](#).

Dil süzgeci

Dosya maskesine göre tarama yapılırken çeviriyi süzmek için kullanılacak kurallı ifade. Weblate tarafından yönetilen dillerin listesini sınırlamak için kullanılabilir.

Not: Dil kodlarını dosya adında gördükleri şekilde listelemeniz gerekir.

Bazı süzme örnekleri:

Süzgeç açıklaması	Kurallı ifade
Yalnızca seçilmiş diller	<code>^(cs de es)\$</code>
Dilleri katmama	<code>^(?! (it fr)\$) .+\$</code>
Yalnızca iki harfli kodları süzme	<code>^[.]+\$</code>
Dil dosyası olmayanları katmama	<code>^(?! (blank)\$) .+\$</code>
Tüm dosyaları katma (varsayılan)	<code>^[^.] +\$</code>

Çeşitler kurallı ifadesi

Bir dizenin değişkenlerini belirlemek için kullanılacak kurallı ifade. Bilgi almak için: `:ref: variants`.

Not: Alanların çoğu, proje sahipleri veya yöneticiler tarafından Weblate arayüzünden düzenlenebilir.

Ayrıca bakınız:

Does Weblate support other VCSes than Git and Mercurial?, alerts

Öncelik

Yüksek öncelikli bileşenler çevirmenlere öncelikli olarak sunulur.

4.15 sürümünde değişti: Bu seçenek artık eşleşen sözlük terimlerinin sıralamasını da etkiliyor.

Kısıtlı erişim

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.

Varsayılan değer `DEFAULT_RESTRICTED_COMPONENT` seçeneği ile değiştirilebilir.

İpucu: Bu durum, proje yöneticileri için de geçerlidir. Lütfen durumu değiştirdikten sonra bileşene erişiminizi kaybetmeyeceğinizden emin olun.

Projelerde paylaşım

You can choose additional projects where the component will be visible. Useful for shared libraries which you use in several projects.

Not: 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.

Sözlük olarak kullanılsın

4.5 sürümünde geldi.

Bu bileşenin sözlük olarak kullanılmasını sağlar. Nasıl listeleneceğini *Sözlük rengi* ile yapılandırabilirsiniz.

The glossary will be accessible in all projects defined by *Projelerde paylaşım*.

It is recommended to enable *Dizge yönetimi* on glossaries in order to allow adding new words to them.

Ayrıca bakınız:

Sözlük

Sözlük rengi

Görüntülenen sözcüğün eşleştiği sözlüğün görüntülenme rengi.

2.7.5 Kalıp kodlaması

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:

- İşleme iletisi biçimlendirmesi. Bilgi almak için: *Bileşen yapılandırması*
- Çeşitli eklentiler
 - *Bileşen keşfi*
 - *İstatistik oluşturma*
 - *Eklentiden betikleri çalıştırma*

Bileşen kalıplarında kullanılabilecek değişkenler şunlardır:

```
{{ language_code }}
```

Dil kodu

```
{{ language_name }}
```

Dil adı

```
{{ component_name }}
```

Bileşen adı

```
{{ component_slug }}
```

Bileşen kısaltması

```
{{ project_name }}
```

Proje adı

```
{{ project_slug }}
```

Proje kısaltması

```
{{ url }}
```

Çeviri adresi

```
{{ filename }}
```

Çeviri dosya adı

```
{{ stats }}
```

Çeviri istatistikleri, ek öznitelikleri vardır. Aşağıda örnekleri bulabilirsiniz.

```
{{ stats.all }}
```

Toplam dizge sayısı

```
{{ stats.fuzzy }}
```

Onaylanmayı bekleyen dizge sayısı

```
{{ stats.fuzzy_percent }}
```

Onaylanmayı bekleyen dizge yüzdesi

```
{{ stats.translated }}
```

Çevrilmiş dizge sayısı

```
{{ stats.translated_percent }}
```

Çevrilmiş dizge yüzdesi

```
{{ stats.allchecks }}
```

Denetim hatası olan dizgelerin sayısı

```
{{ stats.allchecks_percent }}
```

Denetim hatası olan dizgelerin yüzdesi

```
{{ author }}
```

Geçerli işleme sorumlusu. Yalnızca işleme kapsamında kullanılabilir.

`{{ addon_name }}`

Yalnızca eklenti işleme iletisinde kullanılabilen, işlenen geçerli eklentinin adı.

Depo tarayıcısında ya da düzenleyici kalıplarında kullanılabilecek değişkenler:

`{{branch}}`

geçerli dal

`{{line}}`

dosyadaki satır

`{{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 }}
```

...and other Django template features.

2.7.6 İçe aktarma hızı

Fetching VCS repository and importing translations to Weblate can be a lengthy process, depending on size of your translations. Here are some tips:

Yapılandırmayı iyileştirin

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 [Üretim kurulumu](#) for more details, especially:

- Configure Celery for executing background tasks (see [Celery ile arka plan görevlerini kullanmak](#))
- [Ön bellek özelliğini açın](#)
- [Güç bir veri tabanı sunucusu kullanın](#)
- [Hata ayıklama kipini kapatın](#)

Kaynak sınırlarını denetleyin

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 *Celery ile arka plan görevlerini kullanmak*).

Gereksiz denetimleri devre dışı bırakın

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 Bileşenleri kendiliğinden oluşturmak

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 *Bileşen keşfi* 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*.

Ayrıca bakınız:

Yönetim komutları, *Bileşen keşfi*

2.8 Dil tanımları

Farklı çevirileri doğru bir şekilde sunmak için dil adı, metin yönü, çoğul tanımları ve dil kodu hakkındaki bilgiler gereklidir.

2.8.1 Hazır dil tanımları

Definitions for about 600 languages are included in Weblate and the list is extended in every release. Whenever Weblate is upgraded (more specifically whenever **weblate migrate** is executed, see *Generic upgrade instructions*) the database of languages is updated to include all language definitions shipped in Weblate.

This feature can be disabled using *UPDATE_LANGUAGES*. You can also enforce updating the database to match Weblate built-in data using *setuplang*.

Ayrıca bakınız:

İç dil tanımlarını zenginleştirmek, *Geçerli dil tanımları*

2.8.2 Dil kodlarını işlemek

Weblate, çevirileri işlerken, dil kodunu (genellikle ISO 639-1 olanı) *Dosya maskesi* üzerinden var olan herhangi bir dil nesnesiyle eşleştirmeye çalışır.

Dil kod adları ile bu eşleştirmeyi proje düzeyinde daha ayrıntılı ayarlayabilirsiniz.

Tam eşleşme bulunamazsa, var olan dillerden en iyi uyanı ile eşleştirilmeye çalışılır. Şu adımlar denenir:

- Büyük/küçük harfe duyarlı olmadan bakılır.
- Alt çizgiler ve kısa çizgiler normalleştirilir.
- Yerleşik dil kod adlarına bakılır.
- Dil adına göre bakılır.
- Belirli bir dil için varsayılan ülke kodu yok sayılır. `cs_CZ` yerine `cs` seçilir.

Bunun da başarısız olması durumunda, varsayılan ayarlar kullanılarak (soldan sağa metin yönü, bir tane çoğul) yeni bir dil tanımı oluşturulur. `xx_XX` koduyla kendiliğinden oluşturulan dil `xx_XX` (*oluşturuldu*) olarak adlandırılır. Bunu daha sonra yönetici arayüzünde değiştirmek (bilgi almak için: *Dil tanımlarını değiştirmek*) ve sorun izleyiciye bildirmek (bilgi almak için: *Weblate için katkıda bulunmak*) isteyebilirsiniz. Böylece doğru tanımın gelecek Weblate sürümüne eklenmesini sağlayabilirsiniz.

İpucu: Dil olarak istenmeyen bir şey görürseniz, çevirileri işlerken bu türde bir dosyanın yok sayılması için *Dil süzgeci* seçeneğini ayarlamak isteyebilirsiniz.

Ayrıca bakınız:

Dil kodu, Yeni çevirileri eklemek

2.8.3 Dil tanımlarını değiştirmek

You can change language definitions in the languages interface (`/languages/` URL).

While editing, make sure all fields are correct (especially plurals and text direction), otherwise translators will be unable to properly edit those translations.

2.8.4 Belirsiz dil kodları ve makro diller

In many cases it is not a good idea to use macrolanguage code for a translation. The typical problematic case might be Kurdish language, which might be written in Arabic or Latin script, depending on actual variant. To get correct behavior in Weblate, it is recommended to use individual language codes only and avoid macrolanguages.

Ayrıca bakınız:

Makro diller tanımları, Makro dillerin listesi

2.8.5 Dil tanımları

Her dilde şu alanlar bulunur:

Dil kodu

Code identifying the language. Weblate prefers two letter codes as defined by [ISO 639-1](#), but uses [ISO 639-2](#) or [ISO 639-3](#) codes for languages that do not have two letter code. It can also support extended codes as defined by [BCP 47](#).

Ayrıca bakınız:

Dil kodlarını işlemek, Yeni çevirileri eklemek

Dil adı

Visible name of the language. The language names included in Weblate are also being localized depending on user interface language.

Metin yönü

Determines whether language is written right to left or left to right. This property is autodetected correctly for most of the languages.

Çoğul sayısı

Number of plurals used in the language.

Çoğul formülü

Gettext compatible plural formula used to determine which plural form is used for given count.

Ayrıca bakınız:

Çoğullar, GNU gettext araçları: Plural forms, Unikod Konsorsiyumu Dil Çoğul Biçim Kuralları

Konuşan sayısı

Dünyada bu dili konuşan kişi sayısı.

2.8.6 Yeni çevirileri eklemek

2.18 sürümünde değişti: In versions prior to 2.18 the behaviour of adding new translations was file format specific.

Weblate can automatically start new translation for all of the file formats.

Some formats expect to start with an empty file and only translated strings to be included (for example *Android string resources*), while others expect to have all keys present (for example *GNU gettext*). The document-based formats (for example *OpenDocument biçimi*) 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 dosyaları*).

When you specify *Yeni çeviriler kalıbı* in *Bileşen yapılandırması*, Weblate will use this file to start new translations. Any exiting translations will be removed from the file when doing so.

When *Yeni çeviriler kalıbı* is empty and the file format supports it, an empty file is created where new strings will be added once they are translated.

The *Dil kodu biçimi* allows you to customize language code used in generated filenames:

Dosya biçimine göre varsayılan

Dosya biçimine bağlıdır. Çoğu için POSIX kullanılır.

Ayraç olarak alt çizgi kullanan POSIX biçemi

Genellikle gettext ve ilgili araçlar tarafından kullanılır. `pt_BR` gibi dil kodlarını üretir.

Ülke kodu ile birlikte ayraç olarak alt çizgi kullanan POSIX biçemi

Gerekli olmadığında bile ülke kodunu içeren POSIX biçemindeki dil kodu (`tr_TR` gibi).

Ayraç olarak tire kullanan BCP biçemi

Genellikle web platformlarında kullanılır. “pt-BR” gibi dil kodlarını üretir.

Ülke kodu ile birlikte ayraç olarak tire kullanan BCP biçemi

Gerekli olmadığında bile ülke kodunu içeren BCP biçemindeki dil kodu (`tr-TR` gibi).

Ayraç olarak tire kullanan BCP biçemi, eski dil kodları

Çince ve BCP biçeminde gösterim için eski kodları kullanır.

Ayraç olarak tire kullanan BCP biçemi, küçük harfli

BCP biçeminde gösterim. Tümü küçük harfle (örneğin `tr-tr`).

Apple App Store üst veri biçemi

Üst verileri Apple App Store üzerine yüklemeye uygun biçem.

Google Play üst veri biçemi

Üst verileri Google Play Store üzerine yüklemeye uygun biçem.

Android biçemi

Yalnızca Android uygulamalarında kullanıldığında, `tr-TR` gibi dil kodları üretir.

Linux biçemi

Linux tarafından kullanılan yerel ayarlar, Çince ve POSIX biçeminde gösterim için eski kodları kullanır.

Ek olarak, *Dil kod adları* içinde tanımlanan eşlemeler ters yönde uygulanır.

Not: Weblate, çeviri dosyalarını işlerken bunlardan herhangi birini tanır. Yukarıdaki ayarlar yalnızca yeni dosyaların nasıl oluşturulacağını etkiler.

Ayrıca bakınız:

Dil kodu, Dil kod adları, Dil kodlarını işlemek

2.9 Sürekli yerelleştirme

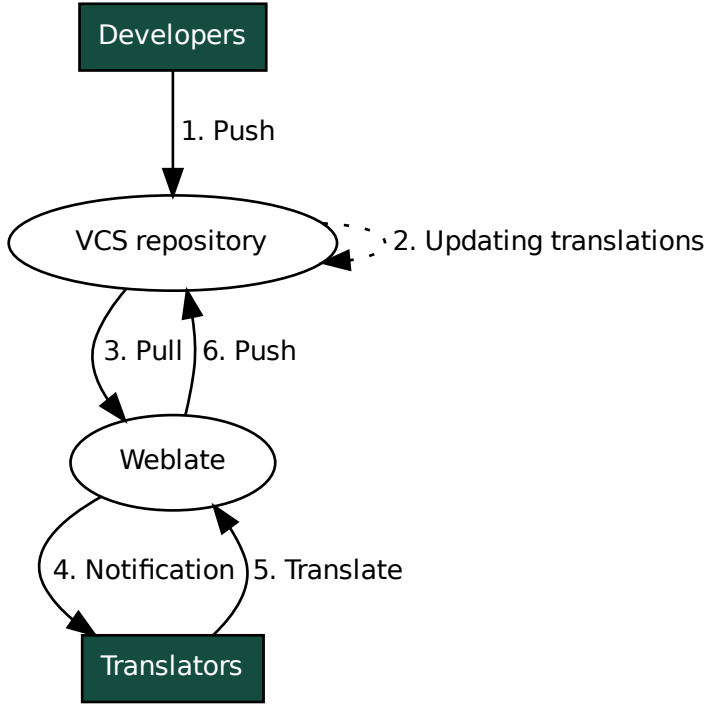
Çevirinizin gelişimi yakından izleyen hazır bir altyapı vardır. Böylece çevirmenler, yayın öncesinde büyük miktarda yeni metinler üzerinde çalışmak yerine, tüm zaman boyunca çeviriler üzerinde çalışabilirler.

Ayrıca bakınız:

`/dev/integration` geliştirme çalışmalarınızı Weblate ile bütünleştirmenin temel yollarını açıklar.

Süreç şu şekildedir:

1. Geliştiriciler değişiklikler yapar ve bunları sürüm denetimi sistemi deposuna gönderir.
2. İsteğe bağlı olarak çeviri dosyaları güncellenir (dosya biçimine bağlıdır, bilgi almak için: *Why does Weblate still show old translation strings when I've updated the template?*).
3. Weblate, sürüm denetimi sistemi deposundan değişiklikleri çeker, bilgi almak için: *Depoları güncellemek*.
4. Weblate çevirilerdeki değişiklikleri algıladığında, çevirmenleri abonelik ayarlarına göre bilgilendirir.
5. Çevirmenler Weblate arayüzünü kullanarak çeviriler yapar ya da çevrimdışı yaptıkları değişiklikleri yükler.
6. Çevirmenlerin çalışması tamamlandıktan sonra, Weblate değişiklikleri yerel depoya gönderir (bilgi almak için: *Lazy commit işlemleri*) ve bunu yapma izni varsa bunları geri iter (bilgi almak için: *Weblate üzerindeki değişiklikleri itmek*).



2.9.1 Depoları güncellemek

Arka uç depolarını kaynaklarından güncellemek için bir yöntem ayarlamalısınız.

- Yaygın kullanılan kod barındırma hizmetlerinin çoğuyla bütünleştirmek için `:ref: hooks'` kullanın:
 - *GitHub değişikliklerini kendiliğinden almak*
 - *GitLab değişikliklerini kendiliğinden almak*
 - *Bitbucket değişikliklerini kendiliğinden almak*
 - *Pagure değişikliklerini kendiliğinden almak*
 - *Azure Repos değişikliklerini kendiliğinden almak*
 - *Gitea depoları değişikliklerini kendiliğinden almak*
- Depo yönetiminden ya da *Weblate REST API uygulaması* ya da *Weblate istemcisi* kullanarak güncellemeyi el ile başlatabilirsiniz
- Weblate kopyanızdaki tüm bileşenlerin kendiliğinden güncellenmesi için `AUTO_UPDATE` seçeneğini etkinleştirin
- `updategit` komutunu yürütün (projeyi seçin ya da tümünü “—güncellemek için `all` kullanın)

Weblate depoyu her güncellediğinde, güncelleme sonrası eklentileri tetiklenir. Bilgi almak için: [Eklentiler](#).

Birleştirme çakışmalarından kaçınmak

Aynı dosya hem Weblate üzerinde hem de Weblate dışında değiştirildiğinde Weblate üzerinden gelen birleştirme-lerde çakışmalar ortaya çıkar. Bu sorunu çözmek için kullanılabilecek iki yaklaşım vardır. Weblate dışındaki düzenlemelerden kaçınmak ya da güncelleme sürecinizi Weblate ile bütünleştirmek. Böylece Weblate dışındaki dosyalar güncellenmeden önce değişiklikler temizlenir.

Tek dilli dosyalar için ilk yaklaşım kolaydır. Weblate üzerinde yeni dizgeler ekleyebilir ve dosyaların tüm düzenleme işlemlerini orada yapabilirsiniz. İki dilli dosyalar için, kaynak kodundan çevrilebilir dosyalar oluşturmak için genellikle bir tür ileti ayıklama işlemi vardır. Bazı durumlarda bu işlem iki bölüme ayrılabilir. Birinci adım ayıklama kalıbını oluşturur (örneğin gettext POT **xgettext** kullanılarak oluşturulur) ve ikinci adım onu gerçek çevirilerle birleştirir (gettext PO dosyaları **msgmerge** kullanılarak güncellenir). İkinci adımı Weblate içinde yapabilirsiniz ve bu işlemten önce bekleyen tüm değişikliklerin katıldığına emin olabilirsiniz.

İkinci yaklaşım, [Weblate REST API uygulaması](#) ile Weblate uygulamasını bekleyen tüm değişiklikleri itmek ve kendi tarafınızda değişiklikler yaparken çeviriyi kilitlemek yoluyla uygulanabilir.

Güncelleme betiği şunun gibi görünebilir:

```
# 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
```

Aynı depoyu paylaşan birden fazla bileşeniniz varsa hepsini ayrı ayrı kilitlemeniz gerekir:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Not: Örnekte, uzaktan Weblate yönetimi için yapılandırmaya (API anahtarları) gerek duyan [Weblate istemcisi](#) kullanılır. Bunu wlc yerine curl gibi herhangi bir HTTP istemcisini kullanarak da yapabilirsiniz. Bilgi almak için: [Weblate REST API uygulaması](#).

Ayrıca bakınız:

[Weblate istemcisi](#)

GitHub değişikliklerini kendiliğinden almak

Weblate doğal GitHub desteği ile gelir.

Hosted Weblate kullanıyorsanız, [Weblate uygulaması](#) kurmanız önerilir. Böylece çok fazla şeyi ayarlamanız gerekmeden doğru kurulumu elde edersiniz. Değişiklikleri geri itmek için de kullanılabilir.

GitHub deposuna yapılan her itmede bildirim almak için, depo ayarlarına (*Webhooks*) aşağıdaki görseldeki gibi Weblate web kancasını ekleyin:

The screenshot shows the GitHub 'Add webhook' configuration page. The left sidebar contains a menu with 'Options', 'Collaborators & teams', 'Branches', 'Webhooks' (highlighted), 'Integrations & services', 'Deploy keys', and 'Alerts'. The main content area is titled 'Webhooks / Add webhook' and includes the following fields and options:

- Payload URL ***: A text input field containing 'https://hosted.weblate.org/hooks/github/'.
- Content type**: A dropdown menu set to 'application/x-www-form-urlencoded'.
- Secret**: An empty text input field.
- SSL verification**: A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' with a red button 'Disable SSL verification'.
- Which events would you like to trigger this webhook?**: Three radio button options:
 - ☒ Just the push event.
 - ☐ Send me everything.
 - ☐ Let me select individual events.
- Active**: A checked checkbox with the text 'We will deliver event details when this hook is triggered.'
- Add webhook**: A green button at the bottom.

The footer of the page shows '© 2018 GitHub, Inc.' and various links like 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

Yük adresi olarak, Weblate adresinizin sonuna “/hooks/github/” ekleyin. Örneğin Hosted Weblate hizmeti için “<https://hosted.weblate.org/hooks/github/>” kullanabilirsiniz.

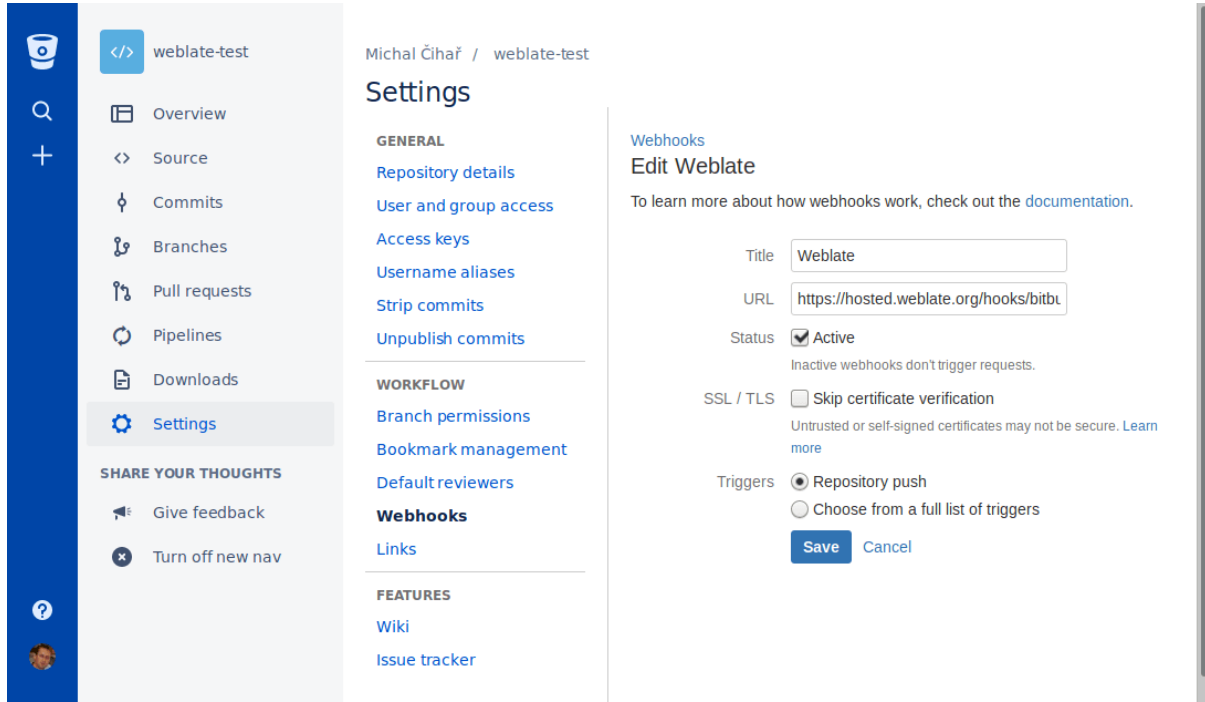
Diğer ayarları varsayılan değerlerinde bırakabilirsiniz (Weblate her iki içerik türünü de işleyebilir ve yalnızca *push* işlemine gerek duyar).

Ayrıca bakınız:

POST /hooks/github/, Hosted Weblate üzerinden depolara erişmek

Bitbucket değişikliklerini kendiliğinden almak

Weblate, Bitbucket web kancalarını destekler. Weblate kurulumunuza hedef olarak `/hooks/bitbucket/` adresiyle depo itme işlemi sırasında tetiklenecek bir web kancası ekleyin (`https://hosted.weblate.org/hooks/bitbucket/` gibi).



Ayrıca bakınız:

`POST /hooks/bitbucket/`, *Hosted Weblate üzerinden depolara erişmek*

GitLab değişikliklerini kendiliğinden almak

Weblate, GitLab kancalarını destekler. Weblate kurulumunuza hedef olarak `/hooks/gitlab/` adresiyle hedefi bir proje web kancası ekleyin (`https://hosted.weblate.org/hooks/gitlab/` gibi).

Ayrıca bakınız:

`POST /hooks/gitlab/`, *Hosted Weblate üzerinden depolara erişmek*

Pagure değişikliklerini kendiliğinden almak

3.3 sürümünde geldi.

Weblate, Pagure kancalarını destekler. Weblate kurulumunuza hedef olarak `/hooks/pagure/` web kancasını ekleyin (`https://hosted.weblate.org/hooks/pagure/` gibi). Bu işlem, *Proje ayarları* bölümündeki *Web kancaları kullanılsın* seçeneği ile yapılabilir:

The screenshot shows the Weblate interface for a project named 'nijel-test'. The top navigation bar includes 'Browse', 'Create', and a user profile icon. Below the project name, there are links for 'New Issue', 'Open PR', 'Fork', and 'Clone'. The main navigation tabs are 'Source', 'Issues', 'Pull Requests', 'Stats', and 'Settings'. The 'Settings' tab is active, showing a sidebar with various settings categories: Project Settings, Project Details, Default Branch, Private Web Hook Key, API Keys, Project Options (selected), Public Notifications, Users & Groups, Deploy Keys, Hooks, Priorities, Roadmap, Close Status, Custom Issue Fields, Reports, Tags, Quick Replies, Regenerate Repos, Give Project, and Delete Project.

The 'Project Options' section contains the following settings:

- ☐ 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

Below these options, there is a section for 'Activate Web-hooks' with a text input field containing 'https://hosted.weblate.org/hooks/pagure/'. Below the input field are two buttons: 'Update' and 'Test web-hook'.

At the bottom, there is a section 'Learn more about' with a list of links:

- Flags
- Tracker read-only
- Pull-request access only
- Roadmap on Issue page
- fedmsg notifications

Ayrıca bakınız:

POST /hooks/pagure/, Hosted Weblate üzerinden depolara erişmek

Azure Repos değişikliklerini kendiliğinden almak

3.8 sürümünde geldi.

Weblate, Azure Repos kancalarını destekler. Weblate kurulumunuza *İtilecek kod* işlemi için hedef olarak `/hooks/azure/` web kancasını ekleyin (`https://hosted.weblate.org/hooks/azure/` gibi). Bu işlem, *Proje ayarları* bölümündeki *Hizmet kancaları* seçeneği ile yapılabilir.

Ayrıca bakınız:

Azure DevOps rehberinde web kancaları, *POST /hooks/azure/, Hosted Weblate üzerinden depolara erişmek*

Gitea depoları değişikliklerini kendiliğinden almak

3.9 sürümünde geldi.

Weblate, Gitea kancalarını destekler. Weblate kurulumunuza *İtme işlemleri* içinden *Gitea web kancası* işlemi için hedef olarak `/hooks/gitea/` web kancasını ekleyin (<https://hosted.weblate.org/hooks/gitea/> gibi). Bu işlem, *Ayarlar* bölümündeki *Web kancaları* seçeneği ile yapılabilir.

Ayrıca bakınız:

Gitea rehberinde web kancaları, *POST /hooks/gitea/*, *Hosted Weblate üzerinden depolara erişmek*

Gitee depoları değişikliklerini kendiliğinden almak

3.9 sürümünde geldi.

Weblate, Gitee kancalarını destekler. Weblate kurulumunuza *İtme* işlemi için hedef olarak `/hooks/gitee/` web kancasını ekleyin (<https://hosted.weblate.org/hooks/gitee/> gibi). Bu işlem, *Yönetim* bölümündeki *Web kancaları* seçeneği ile yapılabilir.

Ayrıca bakınız:

Gitee rehberinde web kancaları, *POST /hooks/gitee/*, *Hosted Weblate üzerinden depolara erişmek*

Depoları her gece kendiliğinden güncellemek

Weblate, daha sonra değişiklik birleştirme başarımını artırmak için her gece uzak depoları kendiliğinden alır. İsteğe bağlı olarak, *AUTO_UPDATE* seçeneğini etkinleştirerek bunu gecelik birleştirmeler yapmaya da dönüştürebilirsiniz.

2.9.2 Weblate üzerindeki değişiklikleri itmek

Her çeviri bileşeni için ayrı bir itme adresi ayarlanabilir (bilgi almak için *Depo itme adresi*) ve bu durumda Weblate, değişikliği uzak depoya itebilir. Weblate, değişiklikleri her işlemede otomatik olarak gönderecek şekilde de yapılandırılabilir (varsayılan davranış, bilgi almak için: *İşleme ile itme*). Değişikliklerin kendiliğinden itilmesini istemiyorsanız, bunu el ile *Depo bakımı* bölümünden ya da *wlc push* API seçeneğini kullanarak yapabilirsiniz.

İtme seçenekleri, kullanılan *Sürüm denetimi bütünleştirmesi* değerine göre farklılık gösterir. Ayrıntılı bilgileri bu bölümden alabilirsiniz.

İtme işleminin doğrudan Weblate tarafından yapılmasını istemiyorsanız, *GitHub çekme istekleri*, *GitLab birleştirme istekleri*, *Gitea çekme isteği*, *Pagure birleştirme istekleri* çekme istekleri ya da *Gerrit* onayları desteklenmektedir. Bunları *Bileşen yapılandırması* içindeki *Sürüm denetimi sistemi* bölümünden *GitHub*, *GitLab*, *Gitea*, *Gerrit* ya da *Pagure* olarak seçerek etkinleştirebilirsiniz.

Genel olarak, Git, GitHub ve GitLab ile şu seçenekler kullanılabilir:

İstenilen kurulum	Sürüm denetimi sistemi	Depo itme adresi	İtme işleminin yapılacağı dal
İtme yok	<i>Git</i>	<i>empty</i>	<i>empty</i>
Doğrudan itme	<i>Git</i>	SSH adresi	<i>empty</i>
Ayrı bir dala it	<i>Git</i>	SSH adresi	Dal adı
GitHub dalından çekme isteği	<i>GitHub çekme istekleri</i>	<i>empty</i>	<i>empty</i>
GitHub dalına itme isteği	<i>GitHub çekme istekleri</i>	SSH URL ¹	Dal adı
GitLab dalından birleştirme isteği	<i>GitLab birleştirme istekleri</i>	<i>empty</i>	<i>empty</i>
GitLab dalından birleştirme isteği	<i>GitLab birleştirme istekleri</i>	SSH URL ^{Sayfa 291, 1}	Dal adı
Gitea çatalından birleştirme isteği	<i>Gitea çekme isteği</i>	<i>empty</i>	<i>empty</i>
Gitea dalından birleştirme isteği	<i>Gitea çekme isteği</i>	SSH URL ¹	Dal adı
Pagure çatalından birleştirme isteği	<i>Pagure birleştirme istekleri</i>	<i>empty</i>	<i>empty</i>
Pagure dalından birleştirme isteği	<i>Pagure birleştirme istekleri</i>	SSH URL ¹	Dal adı

Not: Weblate işledikten sonra değişikliklerin kendiliğinden gönderilmesini de etkinleştirebilirsiniz. Bu işlem *İşleme ile itme* içinden yapılabilir.

Ayrıca bakınız:

SSH anahtarlarını ayarlamak için *Depolara erişmek* ve değişikliklerin Weblate tarafından ne zaman işleneceğine karar verildiği ile ilgili bilgi almak için `:ref:'lazy-commit'` bölümlerine bakabilirsiniz.

Korunmuş dallar

Weblate ile korumalı dal kullanıyorsanız, çekme isteklerini kullanacak ve çeviriler üzerinde gerçek gözden geçirme yapacak bir yapılandırma ayarlayabilirsiniz (bilmediğiniz diller için sorunlu olabilecek şeyler). Alternatif olarak, Weblate itme kullanıcısı için bu sınırlamayı kaldırabilirsiniz.

Örneğin bu işlem GitHub üzerinde, depo yapılandırmasında ayarlanabilir:

¹ *Kaynak kod deposu* itmeyi destekliyorsa boş olabilir.

☒ **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 Diğerleri ile etkileşim

Weblate, API uygulaması -başkalarıyla etkileşim kurmayı kolaylaştırır.

Ayrıca bakınız:

Weblate REST API uygulaması

2.9.4 Lazy commit işlemleri

Weblate, olabiliyorsa aynı yazardan gelen işlemleri tek bir işleme olarak gruplandırarak biçimde davranır. Böylece, işleme sayısı büyük ölçüde azaltılır. Bununla birlikte, sürüm denetimi sistemi deposunu eşitlemek isterseniz bunu açıkça belirtmeniz gerekir. Örneğin birleştirme için (varsayılan olarak *Yöneticiler* grubu için izin verilir, bilgi almak için: `:ref:privileges`).

Bu kipteki değişiklikler, aşağıdaki koşullardan herhangi biri yerine getirildiğinde işlenir:

- Başka biri zaten değiştirilmiş bir dizgeyi değiştirdiğinde.
- Yukarı akıştan bir birleştirme gerçekleştirildiğinde.
- Açık bir işleme isteği yapıldığında.
- Bir dosyanın indirilmesi istendiğinde.
- Değişiklik, *Bileşen yapılandırması* üzerinde *İşlenecek değişikliklerin yaşı* olarak tanımlanmış dönemden daha eski olduğunda.

İpucu: Gönderimler her bileşen için ayrı oluşturulur. Bu nedenle, birçok bileşenin varsa, gene çok sayıda gönderim görürsünüz. Bu durumda *Git işlemlerini bir araya toplar* eklentisini kullanabilirsiniz.

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 *Django yönetim arayüzü*. 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 Betikleri kullanarak depo işlemleri yapmak

Weblate ile depo arasındaki etkileşim *Eklentiler* ile özelleştirilebilir. Eklentiler ile dış betiklerin nasıl yürütüleceği hakkında ayrıntılı bilgi almak için *Eklentiden betikleri çalıştırma* bölümüne bakabilirsiniz.

2.9.6 Bileşenler arasında çevirilerin tutarlılığını sağlamak

Birden çok çeviri bileşeniniz olduğunda, aynı dizgelerin çevirilerinin de aynı olduğundan emin olmak isteyebilirsiniz. Bu tutarlılık birkaç düzeyde sağlanabilir.

Çevirilerin yayılmasını sağlamak

Çevirilerin yayılmasını sağlamak seçeneği etkinken (varsayılan değer nedir, bilgi almak için: *Bileşen yapılandırması*), tüm yeni çeviriler dizgeleri eşleşen tüm bileşenlerde kendiliğinden yapılır. Bu tür çevirilerin kredisi, tüm bileşenlerde çeviri yapan geçerli kullanıcıya verilir.

Not: Çeviri yayılması için, anahtarın tek dilli çeviri biçimleriyle eşleşmesi gerekir. Çeviri anahtarlarını oluştururken bunu aklınızda bulundurun.

Tutarlılık denetimi

Dizgeler farklı olduğunda *Tutarsız* denetimi tetiklenir. Bu tür farklılıkları el ile incelemek ve doğru çeviriyi seçmek için bunu kullanabilirsiniz.

Kendiliğinden çeviri

Farklı bileşenleri temel alan kendiliğinden çeviri, çevirileri bileşenler arasında eşitlemenin bir yöntemi olabilir. El ile tetikleyebilir (bilgi almak için: *Kendiliğinden çeviri*) veya eklentiye kullanarak depo güncellemelerinde kendiliğinden çalışmasını sağlayabilirsiniz (bilgi almak için: *Kendiliğinden çeviri*).

2.10 Çeviri lisanslarını belirlemek

Çevirmenlerin katkıda bulunmak için kabul etmesi gereken lisans koşullarını belirleyebilirsiniz. Bu özellik, herkese açık olan çevirilerin ne amaçla kullanılabileceğini belirlemek özellikle önemlidir.

Bileşen yapılandırması lisans bilgilerini belirtmelisiniz. Yapılabiliyor olsa da, bir katılımcı lisans anlaşmasını zorunlu kılmaktan kaçın.

2.10.1 Lisans bilgileri

Lisans bilgileri (lisans adı ve adresi) belirtildikten sonra, bu bilgiler ilgili *Bileşen yapılandırması* için çeviri bilgileri bölümünde görüntülenir.

Genellikle açık bir onay gerekmiyorsa lisans bilgilerini yayınlamak için en iyi yer burasıdır. Projeniz veya çeviriniz özgür değilse, büyük olasılıkla önceden izin almanız gerekir.

2.10.2 Katılımcı anlaşması

Bir katılımcı lisans anlaşması belirtirseniz, çeviriye yalnızca bunu kabul eden kullanıcılar katkıda bulunabilir. Katılımcı anlaşması çeviriye erişirken açıkça görülebilir:

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	✓						

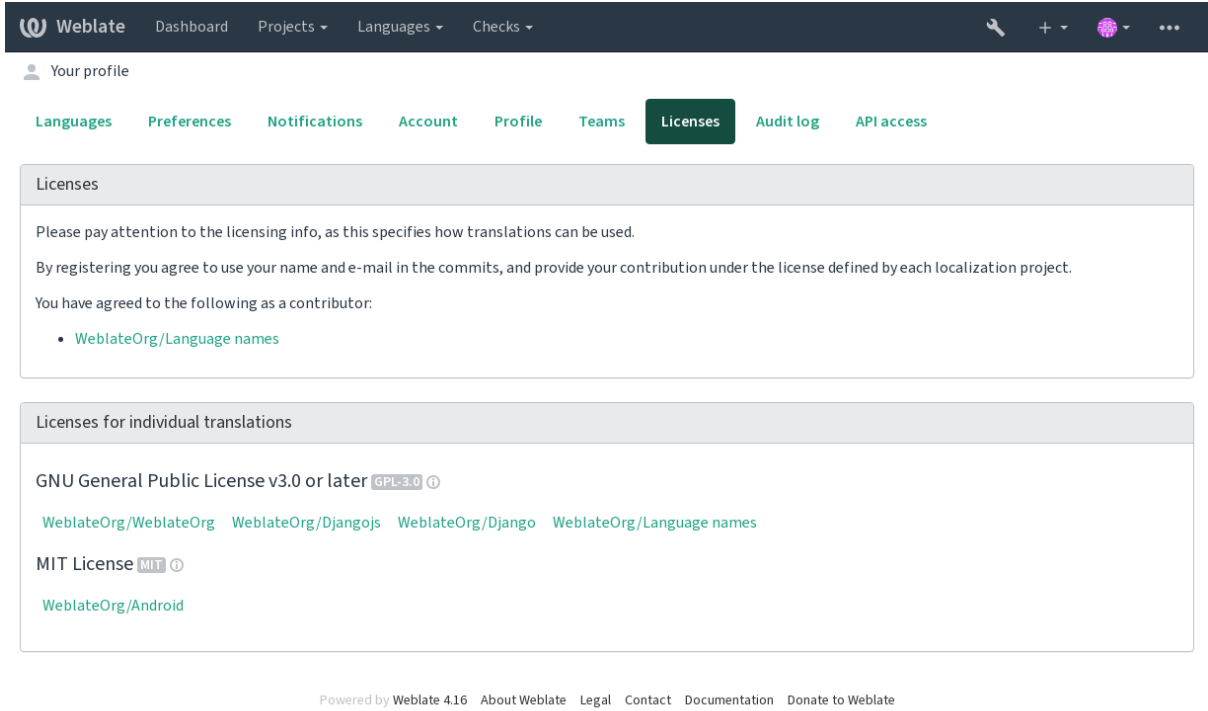
[Start new translation](#)

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

Yazılan metinde paragraflar kullanılabilir ve dış bağlantılar eklenebilir. HTML kod imleri kullanılamaz.

2.10.3 Kullanıcı lisansları

Herhangi bir kullanıcı, bulut sunucusunda herkese açık tüm projelerin tüm çeviri lisanslarını profilinde inceleyebilir:



The screenshot shows the Weblate web application interface. At the top, there 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 highlighted), Audit log, and API access. The 'Licenses' tab is active, 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:'. Below this text is a bulleted list with one item: 'WeblateOrg/Language names'. A second section titled 'Licenses for individual translations' 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). Below these, there are links to 'WeblateOrg/WebplateOrg', 'WeblateOrg/Djangojs', 'WeblateOrg/Django', 'WeblateOrg/Language names', and 'WeblateOrg/Android'. At the bottom of the page, there is a footer with the text 'Powered by Weblate 4.16' and links for About Weblate, Legal, Contact, Documentation, and Donate to Weblate.

2.11 Çeviri süreci

2.11.1 Öneri oylaması

Everyone can add suggestions by default, to be accepted by signed in users. Suggestion voting can be used to make use of a string when more than one signed-in user agrees, by setting up the *Bileşen yapılandırması* with *Suggestion voting* to turn on voting, and *Autoaccept suggestions* to set a threshold for accepted suggestions (this includes a vote from the user making the suggestion if it is cast).

Not: 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 Additional info on source strings

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:

The screenshot shows the Weblate web interface with a modal dialog titled "Edit additional string info" open. The dialog has a close button (X) in the top right corner. It contains the following sections:

- Explanation**: A text input field with a placeholder line. Below it, a description: "Additional explanation to clarify meaning or usage of the string."
- Labels**: Two checkboxes, "Current sprint" (checked) and "Next sprint" (unchecked). Below them, a note: "Additional labels can be defined in the [project settings](#)."
- Translation flags**: A text input field. Below it, a description: "Additional comma-separated flags to influence Weblate behavior."
- Inherited flags**: A list showing "java-printf-format" is inherited from "Android String Resource".

A "Save" button is located at the bottom right of the dialog. The background interface shows the "Source string" section with the key "dow_monday" and its translation "Monday". The right sidebar contains sections like "Things to check", "Variants", "Glossary", "String information", "Screenshot context", "Explanation", "Key", "Labels", "Flags", "String age", "Source string age", and "Translation file".

Access this directly from the translation interface by clicking the “Edit” icon next to *Screenshot context* or *Flags*.

Dizge önceliklendirme

2.0 sürümünde geldi.

String priority can be changed to offer higher priority strings for translation earlier by using the `priority` flag.

İpucu: This can be used to order the flow of translation in a logical manner.

Ayrıca bakınız:

Quality checks

Çeviri işaretleri

2.4 sürümünde geldi.

3.3 sürümünde değişti: Previously called *Quality checks flags*, it no longer configures only checks.

Customization of quality checks and other Weblate behavior, see *Parametreleri kullanarak davranışı özelleştirmek*.

The string flags are also inherited from the *Çeviri işaretleri* at *Bileşen yapılandırması* and flags from the translation file (see *Desteklenen dosya biçimleri*).

Ayrıca bakınız:

Quality checks, Parametreleri kullanarak davranışı özelleştirmek

Açıklama

4.1 sürümünde değişti: 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.

Visual context for strings

2.9 sürümünde geldi.

You can upload a screenshot showing a given source string in use within your program. This helps translators understand where it is used, and how it should be translated.

The uploaded screenshot is shown in the translation context sidebar:

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The top navigation bar shows 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main area is titled 'WeblateOrg / Django / Czech / Translate' and shows a progress bar for 'translated 96%'. The translation process is shown for a string with ID 11/26. The English text is 'Help text for automatic translation tool'. The Czech text is 'Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.' The interface includes buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A 'Glossary' panel on the right shows terms like 'machine translation' and 'project'. A 'String information' panel on the right provides details about the string, including its location, age, and source file.

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

In addition to *Additional info on source strings*, screenshots have a separate management interface under the *Tools* menu. Upload screenshots, assign them to source strings manually, or use optical character recognition to do so.

Once a screenshot is uploaded, this interface handles management and source string association:

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / 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!
One
Orangutan has %d banana.
Other
Orangutan 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 Denetimler ve düzeltmeler

2.12.1 Özel kendiliğinden düzeltmeler

Standart olanların yanında özel kendiliğinden düzeltmeleri de uygulayabilir ve bunları `:setting: AUTOFIX_LIST` içine ekleyebilirsiniz.

Kendiliğinden düzeltmeler güçlüdür, ancak bozulmaya da neden olabilir; Bunları yazarken dikkatli olun.

Örneğin, şu kendiliğinden düzeltme, bir çevirideki `foo` dizgesinin her kopyasını `bar` ile değiştirir:

```
# 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
```

Özel denetimler kurmak için, `AUTOFIX_LIST` içinde Python sınıfının tam yolunu belirtin. Bilgi almak için: `:ref: custom-check-modules`.

2.12.2 Parametreleri kullanarak davranışı özelleştirmek

İmleri kullanarak Weblate davranışını ince ayarlayabilirsiniz. Bu işlem kaynak dizge düzeyinde (bilgi almak için: [Additional info on source strings](#)) ya da *Bileşen yapılandırması* (*Çeviri işaretleri*) içinde yapılabilir. Bazı dosya biçimleri, imlerin doğrudan biçimde belirtilmesini de sağlar (bilgi almak için: [Desteklenen dosya biçimleri](#)).

İmler virgül ile ayrılır. Parametreler iki nokta üst üste ile ayrılır. Dizgeye boşluk veya özel karakterler eklemek için tırnak işaretleri kullanabilirsiniz. Örneğin:

```
placeholders:"special:value":"other value", regex:.*
```

Hem tek hem de çift tırnak işaretleri kabul edilir. Ters bölü kullanılarak özel karakterlerden kaçılır:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

Kullanılabilecek parametrelerin listesi şu şekildedir:

rst-text

Bir metnin reStructuredText belgesi olarak işlenmesini sağlar. *Çeviri değiştirilmemiş* parametresini etkiler.

dos-eol

Unix satır sonu karakterleri yerine DOS karakterlerinin kullanılmasını sağlar (`\n` yerine `\r\n`).

read-only

Dizgenin salt okunur olduğunu ve Weblate üzerinde düzenlenemeyeceğini belirler. Bilgi almak için: *Salt okunur dizgeler*.

priority:N

Dizgenin önceliğini belirler. Yüksek öncelikli dizgeler daha çevrilir. Varsayılan öncelik değeri: 100. Bir dizgenin önceliği ne kadar yüksekse, o kadar önce çevrilmek üzere sunulur.

max-length:N

Bir dizgenin en fazla uzunluğunu N karakter olarak sınırlandırır. Bilgi almak için: [:ref:~check-max-length](#).

xml-text

Metnin XML belgesi olarak işlenmesini sağlar. *XML söz dizimi* ve *XML kodlaması* parametrelerini etkiler.

font-family:NAME

Görselleştirme denetimleri için yazı tipi ailesini belirler. Bilgi almak için: *Yazı tiplerini yönetmek*.

font-weight:WEIGHT

Görselleştirme denetimleri için yazı koyuluğunu belirler. Bilgi almak için: *Yazı tiplerini yönetmek*.

font-size:SIZE

Görselleştirme denetimleri için yazı boyutunu belirler. Bilgi almak için: *Yazı tiplerini yönetmek*.

font-spacing:SPACING

Görselleştirme denetimleri için harf aralığını belirler. Bilgi almak için: [:ref:~fonts](#).

icu-flags:FLAGS

ICU MessageFormat kalite denetiminin davranışını özelleştirecek parametreleri belirler.

icu-tag-prefix:PREFIX

check-java-messageformat kalite denetimi için XML kod imlerinin gerekli olan ön ekini belirler.

placeholders:NAME:NAME2:...

Çeviride beklenen yer belirtici dizgeleri belirler. Bilgi almak için: [:ref:~check-placeholders](#).

replacements:FROM:TO:FROM2:TO2...

Ortaya çıkan metin parametreleri denetlenirken yapılacak değişiklikleri belirler (*Çeviri en fazla uzunluğu* ya da *Çevirinin en fazla uzunluğu* gibi). Tipik kullanım örneği, metnin uzun değerlerle bile sığığından emin olmak için yerleştirilebilir öğelerin genişletilmesidir. Örnek: `replacements:%s:"Ali Kaya"`.

variants:SOURCE

Bu dizgenin, eşleşen kaynağın bir dizge çeşidi olarak imlenmesini sağlar. Bilgi almak için: [:ref:~variants](#).

regex:REGEX

Çevirilerin eşleştirileceği kurallı ifadeyi belirtir. Bilgi almak için: *Kurallı ifade*.

forbidden

Bir sözlükte yasaklanmış bir çeviriyi belirtir. Bilgi almak için: *Yasaklı çeviriler*.

strict-same

“Değiştirilmemiş çeviri” içinde iç kara listedeki sözcüklerin kullanmasını engeller. Bilgi almak için: *Çeviri değiştirilmemiş*.

check-glossary

Sözlük ile uyumlu değil kalite denetimini etkinleştirir.

angularjs-format

AngularJS değer ekleme dizgesi kalite denetimini etkinleştirir.

c-format

C biçimi kalite denetimini etkinleştirir.

c-sharp-format

C# biçimi kalite denetimini etkinleştirir.

es-format

ECMAScript kalıbı sabitleri kalite denetimini etkinleştirir.

i18next-interpolation

i18next değer eklemesi kalite denetimini etkinleştirir.

icu-message-format

ICU MessageFormat kalite denetimini etkinleştirir.

java-printf-format

Java biçimi kalite denetimini etkinleştirir.

java-format

Java MessageFormat kalite denetimini etkinleştirir.

javascript-format

JavaScript biçimi kalite denetimini etkinleştirir.

lua-format

Lua biçimi kalite denetimini etkinleştirir.

object-pascal-format

Object Pascal biçimi kalite denetimini etkinleştirir.

percent-placeholders

Yüzde yer belirticileri kalite denetimini etkinleştirir.

perl-format

Perl biçimi kalite denetimini etkinleştirir.

php-format

PHP biçimi kalite denetimini etkinleştirir.

python-brace-format

Python parantez biçimi kalite denetimini etkinleştirir.

python-format

Python biçimi kalite denetimini etkinleştirir.

qt-format

Qt biçimi kalite denetimini etkinleştirir.

qt-plural-format

Qt çoğul biçimi kalite denetimini etkinleştirir.

ruby-format

Ruby biçimi kalite denetimini etkinleştirir.

scheme-format

Şema biçimi kalite denetimini etkinleştirir.

vue-format

Vue 118n biçimlendirmesi kalite denetimini etkinleştirir.

md-text

Metnin Markdown belgesi olarak işlenmesini sağlar. *Markdown bağlantıları*, *Markdown başvuruları* ve *Markdown söz dizimi* kalite denetimlerini etkinleştirin.

case-insensitive

Denetim davranışını büyük/küçük harfe duyarlı olmayacak şekilde ayarlar. Şu anda yalnızca *Yer belirticiler* kalite denetimini etkiler.

safe-html

Güvenli olmayan HTML kalite denetimini etkinleştirir.

url

Dizge yalnızca bir adresten oluşmalıdır. *Adres* kalite denetimini etkinleştirin.

ignore-all-checks

Tüm kalite denetimlerini yok sayar.

ignore-bbcode

BBCode kodlaması kalite denetimini atlar.

ignore-duplicate

Art arda yinelenen sözcükler kalite denetimini atlar.

ignore-check-glossary

Sözlük ile uyumlu değil kalite denetimini atlar.

ignore-double-space

Çift boşluk kalite denetimini atlar.

ignore-angularjs-format

AngularJS değer ekleme dizgesi kalite denetimini atlar.

ignore-c-format

C biçimi kalite denetimini atlar.

ignore-c-sharp-format

C# biçimi kalite denetimini atlar.

ignore-es-format

ECMAScript kalıbı sabitleri kalite denetimini atlar.

ignore-i18next-interpolation

check-i18next-format kalite denetimini atlar.

ignore-icu-message-format

ICU MessageFormat kalite denetimini atlar.

ignore-java-format

Java MessageFormat kalite denetimini atlar.

ignore-java-printf-format

Java biçimi kalite denetimini atlar.

ignore-javascript-format

JavaScript biçimi kalite denetimini atlar.

ignore-lua-format

Lua biçimi kalite denetimini atlar.

ignore-object-pascal-format

Object Pascal biçimi kalite denetimini atlar.

ignore-percent-placeholders

Yüzde yer belirtecileri kalite denetimini atlar.

ignore-perl-format

Perl biçimi kalite denetimini atlar.

ignore-php-format

PHP biçimi kalite denetimini atlar.

ignore-python-brace-format

Python parantez biçimi kalite denetimini atlar.

ignore-python-format

Python biçimi kalite denetimini atlar.

ignore-qt-format

Qt biçimi kalite denetimini atlar.

ignore-qt-plural-format

Qt çoğul biçimi kalite denetimini atlar.

ignore-ruby-format

Ruby biçimi kalite denetimini atlar.

ignore-scheme-format

Şema biçimi kalite denetimini atlar.

ignore-vue-format

Vue I18n biçimlendirmesi kalite denetimini atlar.

ignore-translated

Çevrilmiş kalite denetimini atlar.

ignore-inconsistent

Tutarsız kalite denetimini atlar.

ignore-kashida

Kashida harfi kullanılmış kalite denetimini atlar.

ignore-md-link

Markdown bağlantıları kalite denetimini atlar.

ignore-md-reflink

Markdown başvuruları kalite denetimini atlar.

ignore-md-syntax

Markdown söz dizimi kalite denetimini atlar.

ignore-max-length

Çevirinin en fazla uzunluğu kalite denetimini atlar.

ignore-max-size

Çeviri en fazla uzunluğu kalite denetimini atlar.

ignore-escaped-newline

\n aynı değil kalite denetimini atlar.

ignore-end-colon

İki nokta üst üste farklı kalite denetimini atlar.

ignore-end-ellipsis

Üç nokta farklı kalite denetimini atlar.

ignore-end-exclamation

Ünlem işareti farklı kalite denetimini atlar.

ignore-end-stop

Nokta işareti farklı kalite denetimini atlar.

ignore-end-question

Soru işareti farklı kalite denetimini atlar.

ignore-end-semicolon

Noktalı virgül farklı kalite denetimini atlar.

ignore-newline-count

Satır sonları farklı kalite denetimini atlar.

ignore-plurals

Çoğul biçim eksik kalite denetimini atlar.

ignore-placeholders

Yer belirticiler kalite denetimini atlar.

ignore-punctuation-spacing

Noktalama aralığı kalite denetimini atlar.

ignore-regex

Kurallı ifade kalite denetimini atlar.

ignore-same-plurals

Tekil ile çoğul biçim aynı kalite denetimini atlar.

ignore-begin-newline

Başta yeni satır var kalite denetimini atlar.

ignore-begin-space

Baştaki boşluk kalite denetimini atlar.

ignore-end-newline

Sonda yeni satır var kalite denetimini atlar.

ignore-end-space

Sondaki boşluk kalite denetimini atlar.

ignore-same

Çeviri değiştirilmemiş kalite denetimini atlar.

ignore-safe-html

Güvenli olmayan HTML kalite denetimini atlar.

ignore-url

Adres kalite denetimini atlar.

ignore-xml-tags

XML kodlaması kalite denetimini atlar.

ignore-xml-invalid

XML söz dizimi kalite denetimini atlar.

ignore-zero-width-space

Sıfır genişlikli boşluk kalite denetimini atlar.

ignore-ellipsis

Üç nokta kalite denetimini atlar.

ignore-icu-message-format-syntax

ICU MessageFormat söz dizimi kalite denetimini atlar.

ignore-long-untranslated

Uzun süredir çevrilmemiş kalite denetimini atlar.

ignore-multiple-failures

Birden çok denetim hatası kalite denetimini atlar.

ignore-unnamed-format

Birden çok değişken adlandırılmamış kalite denetimini atlar.

ignore-optional-plural

Çoğul değil kalite denetimini atlar.

Not: Kural genellikle, belirteci kullanılarak herhangi bir denetim için *ignore-** olarak adlandırılır. Böylece bunu kendi özel denetimleriniz için bile kullanabilirsiniz.

Bu imler hem *Bileşen yapılandırması* ayarlarında, her kaynak dizge ayarı için hem de çeviri dosyasının kendisinde (GNU gettext gibi) anlaşılır.

2.12.3 Denetimleri dayatmak

3.11 sürümünde geldi.

Bileşen yapılandırması içinde *Zorunlu denetimler* ayarını yaparak yok sayılmayacak denetimlerin listesini yapılandırabilirsiniz. Listelenen denetimler kullanıcı arayüzünden kapatılamaz ve bu denetimde hatalı bulunan dizgeler *Düzenlenmesi gerekli* olarak işaretlenir (bilgi almak için: `:ref:states`).

Not: Denetim dayatması kendiliğinden etkinleştirilmez. Denetim, dizge ya da bileşen imlerine karşılık gelen im eklenerek açılabilir.

Ayrıca bakınız:

Additional info on source strings, Çeviri işaretleri

2.12.4 Yazı tiplerini yönetmek

3.7 sürümünde geldi.

İpucu: Weblate üzerine yüklenen yazı tipleri yalnızca *Çeviri en fazla uzunluğu* denetimi için kullanılır. Weblate kullanıcı arayüzünde bir etkisi yoktur.

İşlenen metnin boyutlarını hesaplamak için kullanılan *Çeviri en fazla uzunluğu* denetiminin yazı tipinin Weblate üzerine yüklenmesi ve bir çeviri imi kullanılarak seçilmesi gerekir (bilgi almak için: *Parametreleri kullanarak davranışı özelleştirmek*).

Çeviri projenizin *Yönetim* menüsü altındaki *Yazı tipleri* içindeki Weblate yazı tipi yönetimi aracı, yazı tiplerinin yüklenip yönetilebileceği bir arayüz sağlar. TrueType veya OpenType yazı tipleri yüklenebilir, yazı tipi grupları ayarlanabilir ve bunlar denetimde kullanılabilir.

Yazı tipi grupları, genellikle Latince olmayan farklı diller için gerekli olan farklı yazı tiplerini tanımlanmanızı sağlar:

The screenshot displays the Weblate web interface for managing font groups. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. The breadcrumb trail shows 'WeblateOrg / Font groups / default-font'. The main content area is divided into three sections:

- Font group:** A table showing the 'default-font' group. It lists the 'Default font' as 'Source Sans 3 Bold'. Under the 'Japanese' and 'Korean' columns, there are 'language override' buttons and the font 'Droid Sans Fallback Regular'. Each override has a 'Remove' button. A 'Delete' button is at the bottom left of this section.
- Add language override:** A form with a 'Language' dropdown (currently empty) and a 'Font' dropdown (currently empty). A 'Save' button is at the bottom.
- Edit font group:** A form for editing the 'default-font' group. It has a 'Font group name' field (containing 'default-font') and a 'Default font' dropdown (containing 'Source Sans 3 Bold'). A note states: 'Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.' and 'Default font is used unless per language override matches.' A 'Save' button is at the bottom.

At the very bottom, a footer line reads: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

Yazı tipi grupları, boşluk veya özel karakterler içermeyen bir ad ile tanımlanır. Böylece denetim tanımında kolayca kullanılabilir:

W Weblate

DashboardProjectsLanguagesChecks

⚙️ + 🌐 ⋮

WebOrg / Fonts

Font groupsFonts

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


Yazı tipi ailesi ve biçemi, yüklendikten sonra kendiliğinden tanınır:

W Weblate

DashboardProjectsLanguagesChecks

⚙️ + 🌐 ⋮

WebOrg / 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

Weblate üzerine çeşitli yazı tipleri yükleyebilirsiniz:

Dizge uzunluğunu denetleme üzere yazı tiplerini kullanmak için, uygun imleri aktarın (bilgi almak için: [Parametreleri kullanarak davranışı özelleştirmek](#)). Büyük olasılıkla şunlara gerek duyacaksınız:

max-size:500

En fazla genişliği piksel olarak belirler.

font-family:ubuntu

Belirteci ile kullanılacak yazı tipi grubunu tanımlar.

font-size:22

Yazı boyutunu piksel olarak tanımlar.

2.12.5 Kendi denetimlerinizi oluşturmak

Çok çeşitli kalite denetimleri hazır olarak bulunur (bilgi almak için: [Quality checks](#)). Ancak denetlemek istediğiniz her şey kapsanmamış olabilir. Yapılabilecek denetimlerin listesi [CHECK_LIST](#) kullanılarak ayarlanabilir. Ayrıca kendi özel denetimlerinizi de ekleyebilirsiniz.

1. `weblate.checks.Check` alt sınıfı
2. Birkaç öznelik ayarlayın.
3. `check` (kodunuzdaki çoğulları işlemek istiyorsanız) ya da `check_single` yöntemini uygulayın (sizin için hangisi işe yarıyorsa).

Bazı örnekler:

Özel denetimler kurmak için, [CHECK_LIST](#) içinde Python sınıfının tam yolunu belirtin. Bilgi almak için: `:ref:`custom-check-modules``.

“foo” içermeyen çeviri metinlerini denetlemek

Çeviride “foo” dizgesinin eksik olup olmadığına bakan basit bir denetim.

```
# 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
```

Çekçe çeviri metnlerinde çoğulların farklı olup olmadığı denetlemek

Çekçe iki çoğul biçimin aynı olmadığını doğrulamak için dil bilgilerini kullanan denetim.

```
# 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 targets[1] == targets[2]
        return False
```

(sonraki sayfaya devam)


(önceki sayfadan devam)





```
def check_single(self, source, target, unit):  
    """We don't check target strings here."""  
    return False
```


2.13 Kendiliğinden önerileri yapılandırmak

4.13 sürümünde değişti: Weblate 4.13 sürümünden önce, hizmetler *Yapılandırma* içinden yapılandırılıyordu.

Çeşitli makine çevirisi ve çeviri belleği hizmetlerinin desteği içeriden sağlanır. Her hizmet, yönetici tarafından tüm site için ya da projeye özel olarak ayarlanabilir:


[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.

Not: Hizmetlerin kendi kullanım koşulları olduğundan, bunları istediğiniz gibi kullanmanıza izin verildiğinden emin olun.

Hizmetler, *Bileşen yapılandırması* konumunda yapılandırılmış kaynak dilden çeviri yapar. Bilgi almak için: `:ref:`component-source_language``.

Ayrıca bakınız:

Kendiliğinden öneriler

2.13.1 Amagama

Hizmet kimliği

amagama

Yapılandırma

Bu eklentinin yapılandırması yok.

Virtual geliştiricileri tarafından işletilen özel *tmserver* kurulumu.

Ayrıca bakınız:

Installing amaGama, Amagama, amaGama çeviri belleği

2.13.2 Apertium APy

Hizmet kimliği

apertium-apy

Yapılandırma

url	API adresi
-----	------------

Sınırlı bir dil kümesine çeviri sağlayan bir özgür yazılım makine çevirisi platformu.

Apertium kullanmak için kendi Apertium-APy sunucunuzu işletmeniz önerilir.

Ayrıca bakınız:

Apertium web sitesi, Apertium APy belgeleri

2.13.3 AWS

3.1 sürümünde geldi.

Hizmet kimliği

aws

Yapılandırma

key	Erişim anahtarı kimliği
secret	API gizli anahtarı
region	Bölge adı

Amazon Translate, desteklenen çok sayıda dilde İngilizce'ye ve İngilizce'den metin çevirme için kullanılabilecek bir nöral makine çevirisi hizmetidir.

Ayrıca bakınız:

[Amazon Translate belgeleri](#)

2.13.4 Baidu

3.2 sürümünde geldi.

Hizmet kimliği

baidu

Yapılandırma

key	İstemci kimliği
secret	İstemci parolası

Baidu tarafından sunulan makine çevirisi hizmeti.

Bu hizmet bir API kullanır ve bunu kullanmak için Baidu üzerinden bir kimlik ve API anahtarı edinmeniz gerekir.

Ayrıca bakınız:

[Baidu Translate API](#)

2.13.5 DeepL

2.20 sürümünde geldi.

Hizmet kimliği

deepl

Yapılandırma

url	API adresi
key	API anahtarı

DeepL, birkaç dil için iyi makine çevirisi sağlayan ücretli bir hizmettir. *DeepL API* aboneliği satın almanız gerekir veya eski *DeepL Pro (klasik)* tarifesini kullanabilirsiniz.

DeepL hizmetiyle kullanılacak API adresi. Bu bilgi yazıldığında, API v1 yanında ücretsiz ve ücretli API v2 sürümleri bulunuyordu.

<https://api.deepl.com/v2/> (Weblate içinde varsayılan)

Ücretli olarak API kullanımı içindir ve abonelik yöntemi uygulanır.

<https://api-free.deepl.com/v2/>

Ücretsiz olarak API kullanımı içindir ve abonelik yöntemi uygulanır.

<https://api.deepl.com/v1/>

CAT araçları içindir ve kullanıcı başına abonelik yöntemi uygulanır.

Weblate daha önce, DeepL tarafından bir CAT aracı olarak sınıflandırılmıştı. Bu nedenle API v1 kullanması gerekiyordu. Ancak şimdi API v2 kullanması gerekiyor. Bu nedenle, v2 varsayılan değerdir ve var olan bir CAT aboneliğiniz varsa ve Weblate tarafından bunun kullanılmasını istiyorsanız bu değeri v1 olarak değiştirebilirsiniz.

Hangisini kullanacağınızı öğrenmenin en kolay yolu, tarayıcınızda aşağıdakine benzer bir adres açmaktır:

https://api.deepl.com/v2/translate?text=Hello&target_lang=TR&auth_key=XXX

XXX yerine kimlik doğrulama anahtarınızı (auth_key) yazın. “Bonjour” içeren bir JSON nesnesi alırsanız adresiniz doğrudur. Yoksa, diğer üçünü deneyin.

Weblate, DeepL resmîlik düzeyini destekler ve dile göre eşleşen birini seçer (örneğin, “de@formal” ve “de@informal” seçenekleri vardır).

Ayrıca bakınız:

[DeepL web sitesi](#), [DeepL fiyatları](#), [DeepL API belgeleri](#)

2.13.6 Glosbe

Hizmet kimliği

glosbe

Yapılandırma

Bu eklentinin yapılandırması yok.

Hemen hemen yaşayan her dil için ücretsiz sözlük ve çeviri belleği.

API kullanımı ücretsizdir. Ancak çevirilerin kullanımı, kullanılan veri kaynağının lisansına bağlıdır. Kötüye kullanımı önlemek için bir IP adresinden belirli bir süre içinde yapılabilecek çağrılar sınırlıdır.

Ayrıca bakınız:

[Glosbe web sitesi](#)

2.13.7 Google Translate

Hizmet kimliği

google-translate

Yapılandırma

key	API anahtarı
-----	--------------

Google tarafından sunulan makine çevirisi hizmeti.

Bu hizmet, Google Çeviri API uygulamasını kullanır. Bir API anahtarı edinmeli ve Google API konsolundan fatu-ralamayı etkinleştirmelisiniz.

Ayrıca bakınız:

[Google translate belgeleri](#)

2.13.8 Google Translate API v3

Hizmet kimliği

google-translate-api-v3

Yapılandırma

credentials	Google Çeviri hizmeti hesap bilgileri
project	Google Çeviri projesi
location	Google Çeviri konumu

Google Cloud hizmetleri tarafından sunulan makine çevirisi hizmeti.

Ayrıca bakınız:

[Google Çeviri belgeleri](#), İstemci kitaplıklarını kullanarak bulut hizmetlerinde kimlik doğrulaması, [Google Çeviri projesi oluşturmak](#), [Google Cloud App Engine konumları](#)

2.13.9 LibreTranslate

4.7.1 sürümünde geldi.

Hizmet kimliği

libretranslate

Yapılandırma

url	API adresi
key	API anahtarı

LibreTranslate, ücretsiz ve açık kaynaklı bir makine çevirisi hizmetidir. Herkese açık kopya için bir API anahtarı gerekir. Ancak LibreTranslate hizmetini kendiniz barındırabilirsiniz ve API uygulamasını ücretsiz olarak kullanabileceğini birkaç yansı bulunur.

<https://libretranslate.com/> (resmi herkese açık kopya)

Web sitesi dışında kullanmak için bir API anahtarı gerekir.

Ayrıca bakınız:

[LibreTranslate web sitesi](#), [LibreTranslate deposu](#), [LibreTranslate yansıları](#)

2.13.10 Microsoft Terminology

2.19 sürümünde geldi.

Hizmet kimliği

microsoft-terminology

Yapılandırma

Bu eklentinin yapılandırması yok.

Microsoft Terminoloji hizmeti API uygulaması, Dil portalı üzerinde bulunan terminoloji, tanım ve kullanıcı arabirimi (UI) dizgelerine bir web hizmeti üzerinden programlama ile erişmenizi sağlar.

Ayrıca bakınız:

[Microsoft Terminology hizmeti API uygulaması](#)

2.13.11 Microsoft Translator

2.10 sürümünde geldi.

Hizmet kimliği

microsoft-translator

Yapılandırma

key	API anah- tarı	
ba- se_ı	Uy- gu- lama temel adre- si	Kullanılabilecek seçenekler: api.cognitive.microsofttranslator.com – Küresel (bölgesiz) api-apc.cognitive.microsofttranslator.com – Asya Pasifik api-eur.cognitive.microsofttranslator.com – Avrupa api-nam.cognitive.microsofttranslator.com – Kuzey Ame- rika api.translator.azure.cn – Çin api.cognitive.microsofttranslator.us – Azure ABD hükü- meti bulutu
end- po- int_	Kim- lik doğ- rula- ma hiz- meti adre- si	Bölgesel ya da çoklu hizmet, aşağıdaki bölge alanı kullanılarak belirtilebilir. Kullanılabilecek seçenekler: api.cognitive.microsoft.com – Genel api.cognitive.azure.cn – Çin api.cognitive.microsoft.us – Azure ABD hükümeti bulutu
re- gi- on	Kim- lik doğ- rula- ma hiz- meti böl- gesi	

Microsoft tarafından Azure portalında Cognitive Services altında sunulan makine çevirisi hizmeti.

Weblate, Translator API V3 kullanır.

Translator Text API V2

Translator API V2 ile kullandığınız anahtar, API 3 ile de kullanılabilir.

Translator Text API V3

Azure portalında bir hesap açmanız ve oradan alacağınız anahtar kullanmanız gerekir. Yeni Azure anahtarlarıyla, `region` seçeneği için hizmetinizin yerel ayarını belirtmeniz gerekir.

İpucu: Azure Çin için lütfen Azure Portal üzerindeki uç noktanızı kullanın.

Ayrıca bakınız:

Cognitive Services - Text Translation API, Microsoft Azure Portal, Temel adresler, "Çok hizmetli bir kaynak ile kimlik doğrulamak" "Erişim kodu ile kimlik doğrulamak" bölümü

2.13.12 ModernMT

4.2 sürümünde geldi.

Hizmet kimliği

modernmt

Yapılandırma

url	API adresi
key	API anahtarı

Ayrıca bakınız:

[ModernMT API](#),

2.13.13 MyMemory

Hizmet kimliği

mymemory

Yapılandırma

email	İletişim e-posta adresi
username	Kullanıcı adı
key	API anahtarı

Makine çevirisi ile çok büyük bir çeviri belleği.

Ücretsiz, anonim kullanım şu anda günde 100 istek ya da `email` içine bir iletişim e-posta adresi yazdığınızda günde 1.000 istek ile sınırlıdır. Ayrıca onlara yazarak daha fazlasını da isteyebilirsiniz.

Ayrıca bakınız:

[MyMemory web sitesi](#)

2.13.14 Netease Sight

3.3 sürümünde geldi.

Hizmet kimliği

netease-sight

Yapılandırma

key	İstemci kimliği
secret	İstemci parolası

NetEase tarafından sunulan makine çevirisi hizmeti.

Bu hizmet bir API kullanır ve bir NetEase anahtarı ile parolası edinmeniz gerekir.

Ayrıca bakınız:

[NetEase Sight çeviri platformu](#)

2.13.15 SAP Translation Hub

Hizmet kimliği

sap-translation-hub

Yapılandırma

url	API adresi
key	API anahtarı
user-name	SAP kullanıcı adı
password	SAP parolası
enable_mt	Makine çevirisi kullanılıp-sın
do-main	Çeviri etki alanı
	Bir çeviri etki alanının kimliği. Örnek: BC. Bir etki alanı belirtmezseniz, yöntem kullanılabilir tüm etki alanlarındaki çevirileri arar.

SAP tarafından sunulan makine çevirisi hizmeti.

Bu hizmeti kullanmak için bir SAP hesabınızın (ve SAP Cloud Platform üzerinde etkinleştirilmiş SAP Translation Hub) olması gerekir.

Veri tabanı terimine ek olarak, makine çevirisi hizmetlerinin de kullanılıp kullanılmayacağını yapılandırabilirsiniz.

Not: Sandbox API erişimi için url ve key değerlerini ayarlamalısınız.

Üretken API erişimi için url, username ve password seçeneklerini ayarlamalısınız.

Ayrıca bakınız:

SAP Translation Hub API, [SAP Translation Hub temel adresini oluşturmak<https://help.sap.com/docs/SAP_TRANSLATION_HUB/ed6ce7a29bdd42169f5f0d7868bce6eb/3a011fba82644259a2cc3>](https://help.sap.com/docs/SAP_TRANSLATION_HUB/ed6ce7a29bdd42169f5f0d7868bce6eb/3a011fba82644259a2cc3)

2.13.16 tmserver

Hizmet kimliği

tmserver

Yapılandırma

url	API adresi
-----	------------

Translate-toolkit ile birlikte geleni kullanarak kendi çeviri belleği sunucunuzu işletebilir ve Weblate ile konuşmasına izin verebilirsiniz. Ayrıca, geliştirilmiş bir tmserver sürümü olan bir amaGama sunucusu ile de kullanabilirsiniz.

1. First you will want to import some data to the translation memory:

```
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. Start tmserver to listen to your requests:

```
tmserver -d /var/lib/tm/db
```

3. Configure Weblate to talk to it, the default URL is `http://localhost:8888/tmserver/`.

Ayrıca bakınız:

[tmserver](#) [Installing amaGama](#), [Amagama](#), [Amagama çeviri belleği](#)

2.13.17 IBM Watson Language Translator

Hizmet kimliği

`ibm`

Yapılandırma

<code>url</code>	API adresi
<code>key</code>	API anahtarı

IBM Watson Language Translator translates text from one language to another. The service offers multiple domain-specific models.

Ayrıca bakınız:

[Watson Language Translator](#), [IBM Cloud API Docs](#)

2.13.18 Weblate

Hizmet kimliği

`weblate`

Yapılandırma

Bu eklentinin yapılandırması yok.

Weblate machine translation service can provide translations for strings that are already translated inside Weblate. It looks for exact matches in the existing strings.

2.13.19 Weblate Çeviri Belleği

2.20 sürümünde geldi.

Hizmet kimliği

`weblate-translation-memory`

Yapılandırma

Bu eklentinin yapılandırması yok.

Use [Çeviri belleği](#) as a machine translation service. Any string that has been translated in past (or uploaded to the translation memory) can be translated in this way.

2.13.20 Yandex

Hizmet kimliği

`yandex`

Yapılandırma

<code>key</code>	API anahtarı
------------------	--------------

Machine translation service provided by Yandex.

This service uses a Translation API, and you need to obtain an API key from Yandex.

Ayrıca bakınız:

[Yandex Translate API](#), Powered by [Yandex.Translate](#)

2.13.21 Youdao Zhiyun

3.2 sürümünde geldi.

Hizmet kimliği

youdao-zhiyun

Yapılandırma

key	İstemci kimliği
secret	İstemci parolası

Machine translation service provided by Youdao.

This service uses an API, and you need to obtain an ID and an API key from Youdao.

Ayrıca bakınız:

[Youdao Zhiyun Natural Language Translation Service](#)

2.13.22 Özel makine çevirisi

You can also implement your own machine translation services using a few lines of Python code. This example implements machine translation in a fixed list of languages using dictionary Python module:

```
# 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,
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
threshold: int = 75,
):
    """Return tuple with translations."""
    for t in dictionary.translate(text):
        yield {"text": t, "quality": 100, "service": self.name, "source": text}
```


You can list your own class in `WEBLATE_MACHINERY` and Weblate will start using that.

2.14 Eklentiler

2.19 sürümünde geldi.

Eklentiler, çeviri iş akışını özelleştirmesi ve kendiliğinden yapılması için yöntemler sağlar. Yöneticiler, her bir çeviri bileşenine *Yönetim* ↓ *Eklentiler* menüsünden eklentiler ekleyip yönetebilir.

İpucu: Eklentileri `API`, `DEFAULT_ADDONS` veya `:d jadmin:install_addon` kullanarak da yapılandırabilirsiniz.



[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 Hazır eklentiler

Kendiliğinden çeviri

3.9 sürümünde geldi.

Eklenti kimliği

`weblate.autotranslate.autotranslate`

Yapılandırma

<code>mo-</code>	Kendi-	Kullanılabilecek seçenekler:
<code>de</code>	liğinden	<code>suggest</code> – Öneri olarak ekle
	çeviri	<code>translate</code> – Çeviri olarak ekle
	kipi	<code>fuzzy</code> – “Düzenlenmesi gerekli” olarak ekle
<code>fil-</code>	Arama	Lütfen tüm dizgeleri çevirdiğinizde var olan tüm çevirileri yok sayılacağını
<code>ter_</code>	süzgeci	unutmayın.
		Kullanılabilecek seçenekler:
		<code>all</code> – Tüm dizgeler
		<code>nottranslated</code> – Çevrilmemiş dizgeler
		<code>todo</code> – Tamamlanmamış dizgeler
		<code>fuzzy</code> – Düzenlenecek olarak imlenmiş dizgeler
		<code>check:inconsistent</code> – Denetim hatası: Tutarsız
<code>autoc</code>	Kendi-	Kullanılabilecek seçenekler:
	liğinden	<code>others</code> – Diğer çeviri bileşenleri
	yapılmış	<code>mt</code> – Makine çevirisi
	çevi-	
	rilerin	
	kaynağı	
<code>com-</code>	Bileşen-	Kaynak olarak kullanılacak bir bileşenin kısaltmasını yazın. Geçerli proje-
<code>po-</code>	ler	deki tüm bileşenlerin kullanılması için boş bırakın.
<code>nent</code>		
<code>en-</code>	Makine	
<code>gi-</code>	çevirisi	
<code>nes</code>	hizmetle-	
	ri	
<code>th-</code>	Skor eşiği	
<code>res-</code>		
<code>holc</code>		

Tetikleyiciler

bileşen güncellemesi, günlük

Makine çevirisini ya da diğer bileşenleri kullanarak dizgeleri kendiliğinden çevirir.

Ne zaman tetiklenir:

- Bileşende yeni dizgeler görüldüğünde.
- Her bileşen için ayda bir kez. `:setting:BACKGROUND_TASKS` kullanılarak yapılandırılabilir.

Ayrıca bakınız:

Kendiliğinden çeviri, Bileşenler arasında çevirilerin tutarlılığını sağlamak

JavaScript yerelleştirme CDN

4.2 sürümünde geldi.

Eklenti kimliği

`weblate.cdn.cdnjs`

Yapılandırma

thres- hold	Çeviri eşiği	Çevirilerin katılacağı eşik değeri.
css_sele	CSS seçici	Yerelleştirilebilir öğeleri algılamak için CSS seçici.
co- okie_nam	Dil tanımlama bilgisi- nin adı	Dil seçiminin kaydedildiği tanımlama bilgisinin adı.
files	Dizgeleri HTML dos- yalarından ayıklama	Geçerli depoda ya da uzak adreslerde çevrilebilir diz- geler için işlenecek dosya adlarının listesi.

Tetikleyiciler

günlük, depoya işlendikten sonra, depo güncellendikten sonra

JavaScript ya da HTML yerelleştirmesinde kullanılmak üzere içerik dağıtım ağına çevirileri yayınlar.

Statik HTML sayfalarını yerelleştirmek veya JavaScript kodunda yerelleştirmeyi yüklemek için kullanılabilir.

Bileşeniniz için, bunları yerelleştirmek amacıyla HTML sayfalarına ekleyebileceğiniz benzersiz bir adres oluşturur. Bilgi almak için `weblate-cdn`.

Ayrıca bakınız:

`cdn-addon-config`, `weblate-cdn`, `cdn-addon-extract`, `cdn-addon-html`

Boş dizgeleri silme

4.4 sürümünde geldi.

Eklenti kimliği

`weblate.cleanup.blank`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlendikten sonra, depo güncellendikten sonra

Çeviri dosyalarından çevirisi olmayan dizgeleri kaldırır.

Çeviri dosyalarında boş dizgelerin görünmemesi için bunu kullanın (yerelleştirme kitaplığınızın kaynak dizgeyi kullanmak yerine boş dizgeler görüntülenmesi gibi durumlarda).

Ayrıca bakınız:

Does Weblate update translation files besides translations?

Çeviri dosyalarını temizleme

Eklenti kimliği

`weblate.cleanup.generic`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlenmeden önce, depo güncellendikten sonra

Tüm çeviri dosyalarını tek dilli temel dosyayla eşleştirecek şekilde günceller. Bu işlem çoğu dosya biçiminde temel dosyada artık bulunmayan eski çeviri anahtarlarını kaldırır.

Ayrıca bakınız:

Does Weblate update translation files besides translations?

Eksik dilleri ekleme

Eklenti kimliği

`weblate.consistency.languages`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

günlük, depoya eklenmesinden sonra

Proje içindeki tüm bileşenler için tutarlı bir dil kümesinin kullanılmasını sağlar.

Eksik diller her 24 saatte bir ve Weblate üzerine yeni diller eklendiğinde denetlenir.

Diğerlerinden farklı olarak, bu eklenti tüm projeyi etkiler.

İpucu: Yeni eklenen dizgeler *Kendiliğinden çeviri* ile kendiliğinden çevrilir.

Bileşen keşfi

Eklenti kimliği

weblate.discovery.discovery

Yapılandırma

match	Çeviri dosyalarıyla eşleştirilecek kurallı ifade	
file_format	Dosya biçimi	
name_template	Bileşen adını özelleştirme	
base_file_type	Tek dil temel dosya adını tanımlama	İki dilli çeviri dosyaları için boş bırakın.
new_base_name	Yeni çeviriler için temel dosyayı tanımlama	Yeni çevirileri oluşturmak için kullanılan dosyanın adı. Gettext için .pot dosyası olarak seçin.
intermediate_template	Ara dil dosyası	Ara çeviri dosyasının adı. Çoğu durumda bu dosya geliştiriciler tarafından sunulan bir çeviri dosyasıdır ve gerçek kaynak dizgeleri oluşturulurken kullanılır.
language_regex	Dil süzgeci	Dosya maskesi taranırken çeviri dosyalarını süzmek için kullanılan kurallı ifade.
copy_addons	Ana bileşendeki eklentileri yeni oluşturulanlara kopyalama	
remove	Var olmayan dosyalar için bileşenleri kaldırma	
confirm	Yukarıdaki eşleşmelerin doğru görüldüğünü onaylıyorum	

Tetikleyiciler

depo güncellenmesinden sonra

Sürüm denetim sistemindeki dosya değişikliklerine göre proje bileşenlerini kendiliğinden ekler ya da siler.

Sürüm denetimi sistemi her güncellendiğinde tetiklenir ve `:d jadmin:import_project` yönetim komutuna benzer. Bu şekilde, bir sürüm denetimi sistemi içinde birden çok çeviri bileşenini izleyebilirsiniz.

Eşleştirme, karmaşık yapılandırmayı etkinleştiren kurallı ifadeler kullanılarak yapılır. Ancak bunu yapmak için bazı bilgiler gerekir. Yaygın kullanım örnekleri eklenti yardımı bölümünde bulunabilir.

Kaydet üzerine tıkladığınızda, yapılandırmanın gereksinimlerinize uyup uymadığını denetleyebileceğiniz biçimde eşleşen bileşenlerin ön izlemesi sunulur:

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

328

Bölüm 2. Yönetici belgeleri

İpucu: Bileşen bulma eklentisi *İç Weblate adresleri* kullanır. Sürüm denetimi sistemi kurulumunu birden çok bileşen arasında paylaşmanın kullanışlı bir yoludur. Bağlı bileşenler, ilgili her bir bileşenin *Kaynak kod deposu* alanına (Yönetim ↓ Ayarlar ↓ Sürüm denetimi sistemi) `weblate://project/main-component` doldurularak kurulan ana bileşenin yerel deposunu kullanır. Böylece yapılandırma ve sistem kaynakları açısından da zaman kazanılır.

Ayrıca bakınız:

Kalıp kodlaması

Toplu düzenleme

3.11 sürümünde geldi.

Ekenti kimliği

`weblate.flags.bulk`

Yapılandırma

q	Sorgu	
state	Ayarlanacak durum	Kullanılabilecek seçenekler: -1 – Değiştirilmesin 10 – Düzenlenmesi gerekli 20 – Çevrilmiş 30 – Onaylanmış
add_flag:	Eklenenecek çeviri imleri	
remove_flags	Kaldırılacak çeviri imleri	
add_label:	Eklenenecek etiketler	
remove_label:	Kaldırılacak etiketler	

Tetikleyiciler

bileşen güncellemesi

Dizgelerin imlerini, etiketlerini veya durumlarını toplu düzenleyin.

NOT `has:label` arama sorgusuyla başlayarak kendiliğinden etiketleme yapın ve tüm dizgelere gerekli tüm etiketler eklenene kadar sürdürün. Weblate üst verileri için başka kendiliğinden işlemler de yapılabilir.

Örnekler:

Tablo 5: Yeni dizgeler kendiliğinden etiketlenir

Arama sorgusu	<code>NOT has:label</code>
Eklenenecek etiketler	<code>son</code>

Tablo 6: Tüm Uygulama mağazası üst veri dosyaları değişiklik günlüğü dizgelerini salt okunur olarak imleme

Arama sorgusu	<code>language:en AND key:changelogs/</code>
Eklenenecek çeviri imleri	<code>read-only</code>

Ayrıca bakınız:

Toplu düzenleme, *Parametreleri kullanarak davranışı özelleştirmek*, *labels*

Değiştirilmemiş çevirileri “Düzenlenmesi gerekli” olarak imleme

3.1 sürümünde geldi.

Eklenti kimliği

`weblate.flags.same_edit`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

birim oluşturulduktan sonra

Sürüm denetimi sisteminden yeni bir çevrilebilir dizge içe aktarıldığında ve bir kaynak dizgeyle eşleştğinde, Weblate üzerinde düzenlenmesi gerekli olarak işaretlenir. Özellikle çevrilmemiş dizgeler için kaynak dizgeleri içeren dosya biçimleri için kullanışlıdır.

İpucu: Ayrıca, *Çeviri işaretleri* için `strict-same` imini ekleyerek *Çeviri değiştirilmemiş* denetimini sıkılaştırmak isteyebilirsiniz.

Ayrıca bakınız:

Çeviri durumları

Yeni kaynak dizgeleri “Düzenlenmesi gerekli” olarak imleme

Eklenti kimliği

`weblate.flags.source_edit`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

birim oluşturulduktan sonra

Sürüm denetimi sisteminden yeni bir kaynak dizge içe her aktarıldığında, Weblate üzerinde düzenlenmesi gerekli olarak işaretlenir. Böylece geliştiriciler tarafından yazılan kaynak dizgeleri kolayca süzülür ve düzenleyebilirsiniz.

Ayrıca bakınız:

Çeviri durumları

Yeni çevirileri “Düzenlenmesi gerekli” olarak imleme

Eklenti kimliği

`weblate.flags.target_edit`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

birim oluşturulduktan sonra

Sürüm denetimi sisteminden yeni bir çevrilebilir dizge içeri her aktarıldığında, Weblate üzerinde düzenlenmesi gerekli olarak işaretlenir. Böylece geliştiricilerin oluşturduğu çevirileri kolayca süzülür ve düzenleyebilirsiniz.

Ayrıca bakınız:

Çeviri durumları

İstatistik oluşturma

Eklenti kimliği

`weblate.generate.generate`

Yapılandırma

<code>filename</code>	Oluşturulacak dosyanın adı
<code>template</code>	Oluşturulacak dosyanın içeriği

Tetikleyiciler

depoya işlenmeden önce

Çeviri durumu hakkında ayrıntılı bilgiler içeren bir dosya oluşturur.

Hem dosya adında hem de içerikte bir Django kalıbı kullanabilirsiniz. Markup hakkında bilgi almak için [Kalıp kodlaması](#) bölümüne bakabilirsiniz.

Örneğin, her çeviri için bir özet dosyası oluşturma:

Oluşturulacak dosyanın adı

`locale/{{ language_code }}.json`

İçerik

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}"
}
```

Ayrıca bakınız:

[Kalıp kodlaması](#)

Çeviriyi kaynak dizgeyle önceden doldurma

4.11 sürümünde geldi.

Eklenti kimliği

`weblate.generate.prefill`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

bileşen güncellemesi, günlük

Çeviri dizgelerini kaynak dizgeler ile doldurur.

Bileşendeki tüm çevrilmemiş dizgeler kaynak dizgeyle doldurulur ve düzenlenmesi gerekli olarak imlenir. Çeviri dosyalarında boş dizge kabul edilmediğinde bunu kullanın.

Sözde yerel oluşturma

4.5 sürümünde geldi.

Ekenti kimliği

`weblate.generate.pseudolocale`

Yapılandırma

<code>source</code>	Kaynak dizgeler	
<code>target</code>	Hedef çeviri	Bu çevirideki tüm dizgelerin üzerine yazılacak
<code>prefix</code>	Sabit dizge ön eki	
<code>var_prefix</code>	Değişken dizge ön eki	
<code>suffix</code>	Sabit dizge son eki	
<code>var_suffix</code>	Değişken dizge son eki	
<code>var_multipli</code>	Değişken bölüm çarpanı	Kaynak dizgenin uzunluğuna bağlı olarak değişken bölümün kaç kez yinleneceği.
<code>include_readonly</code>	Salt okunur dizgeleri katma	

Tetikleyiciler

bileşen güncellemesi, günlük

Kaynak dizgelere kendiliğinden ön ek ve son ek ekleyerek çeviri oluşturur.

Sözde yereller, yerelleştirme için hazırlanmamış dizgeleri bulmaya yarar. Böylece, uygulamayı sözde yerel dilde çalıştırırken değiştirilmemiş dizgelerin bulunmasını kolaylaştırmak için tüm çevrilebilir kaynak dizgeleri değiştirilir.

Yerelleştirilmiş karşılıkları görünümüne uymayabilecek dizgeler de bulunabilir.

Değişken bölümlerin kullanılması, yerelleştirmeden sonra kullanıcı arabirimine sığmayabilecek dizgelerin bulunmasını sağlar. Metin kaynak dizge uzunluğuna göre genişletilir. Değişken bölümler, metnin uzunluğunun çarpanla çarpılmasıyla yinelenir. Örneğin, değişken son eki `_` olan `Merhaba dünya` ve değişken çarpanı `1` olduğunda `Merhaba dünya_____` olur. Son ek kaynak dizgedeki her karakter için bir kez yinelenir.

Dizgeler şu model kullanılarak oluşturulacak:

Sabit dizge ön eki Değişken dizge ön eki Kaynak dizge Değişken dizge son eki Sabit dizge son eki

İpucu: Deneme için gerçek dilleri kullanabilirsiniz. Bunun yanında Weblate üzerinde *en_XA* ve *ar_XB* gibi bu iş için ayrılmış sahte ayarlar vardır.

İpucu: Bu eklentiye, var olan bir dilin ya da benzer bir dilin yeni bir yerel ayarına çeviri başlatmak için kullanabilirsiniz. Çeviriyi bileşene ekledikten sonra, eklentiye izleyin. *Örnek:* *fr* diliniz varsa ve *fr_CA* çevirisini başlatmak istiyorsanız, kaynak olarak *fr*, hedef olarak *fr_CA* ayarlayın ve ön ek ile son eki boş bırakın.

Kopyalamadan sonra yapılan çevirilerin Weblate tarafından değiştirilmesini önlemek için yeni çeviriyi doldurduktan sonra eklentiye kaldırın.

Katılımcıları yorum bölümünde görüntüleme

Eklenti kimliği

weblate.gettext.authors

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlenmeden önce

PO dosyası üst bilgilerinin yorum bölümü, katılımcı adlarını ve katkı yıllarını içerecek biçimde güncellenir.

PO dosya üst bilgileri şöyle görünür:

```
# 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.
```

“configure” dosyasındaki ALL_LINGUAS değişkeni güncelleme

Eklenti kimliği

weblate.gettext.configure

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya eklendikten sonra, günlük

Yeni bir çeviri eklendiğinde configure, configure.in ya da herhangi bir configure.ac dosyasında ALL_LINGUAS değişkeni güncellenir.

Gettext çıktısını özelleştirme

Eklenti kimliği

weblate.gettext.customize

Yapılandırma

widt	Uzun	Varsayılan olarak gettext satırları 77 karakterden sonra ve yeni satır karakterlerinde kaydırır. –no-wrap parametresi kullanıldığında kaydırma yalnızca yeni satır karakterlerinde yapılır.
kay-	lar	Kullanılabilecek seçenekler:
dır-	ma	“77” – Satırlar, 77 karakterden sonra ve yeni satır karakterlerinde kaydırılır (xgettext varsayılanı)
		65535 – Satırlar yalnızca yeni satırlar karakterlerinde kaydırılır (‘xgettext –no-wrap’ gibi)
		–1 – Satırlar kaydırılmaz

Tetikleyiciler

depolamaya yüklendikten sonra

Gettext çıktısında satır kaydırma gibi davranışların özelleştirilmesini sağlar.

Şu seçenekleri sunar:

- Satırlar 77 karakter ve yeni satır karakterlerinde kaydırılır
- Satırları yalnızca yeni satır karakterlerinde kaydırılır

- Satırlar kaydırılmaz

Not: Varsayılan olarak gettext satırları 77 karakterde ve yeni satır karakterlerinde kaydırır. `--no-wrap` parametresi kullanıldığında kaydırma yalnızca yeni satır karakterlerinde yapılır.

LINGUAS dosyasını güncelleme

Eklenti kimliği

`weblate.gettext.linguas`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya eklendikten sonra, günlük

Yeni bir çeviri eklendiğinde LINGUAS dosyası güncellenir.

MO dosyaları üretme

Eklenti kimliği

`weblate.gettext.mo`

Yapılandırma

<code>path</code>	Oluşturulan MO dosyasının yolu	Belirtilmemişse, PO dosyasının konumu kullanılır.
-------------------	--------------------------------	---

Tetikleyiciler

depoya işlenmeden önce

PO dosyası her değiştirildiğinde MO dosyası kendiliğinden oluşturulur.

Oluşturulan MO dosyasının konumu özelleştirilebilir ve bunun için *Kalıp kodlaması* alanı kullanılır.

PO dosyalarını POT ile eşleştirerek güncelleme (msgmerge)

Eklenti kimliği

`weblate.gettext.msgmerge`

Yapılandırma

<code>previous</code>	Çevrilmiş dizgelerin önceki msgid değerleri saklansın
<code>no_location</code>	Çevrilmiş dizgelerin konumları kaldırılsın
<code>fuzzy</code>	Belirsiz eşleşme kullanılsın

Tetikleyiciler

depo güncellenmesinden sonra

Tüm PO dosyalarını (*Dosya maskesi* ile yapılandırıldığı gibi) **msgmerge** komutunu kullanarak POT dosyasıyla eşleşecek şekilde (*Yeni çeviriler kalıbı* ile yapılandırıldığı gibi) günceller.

Yukarı akış deposundan yeni değişiklikler çekildiğinde tetiklenir. Eklenti yapılandırması aracılığıyla msgmerge komut satırı seçeneklerinin çoğu yapılandırılabilir.

Ayrıca bakınız:

Does Weblate update translation files besides translations?

Git işlemlerini bir araya toplar

Eklenti kimliği

weblate.git.squash

Yapılandırma

squash	İşlemleri bir araya toplama	Kullanılabilecek seçenekler: all – Tüm işlemler bir arada yapılır language – Her dilin işlemleri ayrı yapılır file – Her dosyanın işlemleri ayrı yapılır author – Her katkıda bulunanın işlemleri ayrı yapılır
ap- pend_t	Bir araya toplanmış işlemlerin iletisine art bilgileri eklensin	‘Co-authored-by: ...’ gibi bir işleme iletisinin başka bir serbest biçim bölümünün sonunda, art bilgi satırları RFC 822 e-posta üst bilgilerine benzeyen satırlardır.
com- mit_m	İşleme iletisi	Bu işleme iletisi, bir araya toplanmış işleme iletilerinden gelen birleşik işleme iletileri yerine kullanılır.

Tetikleyiciler

depoya işlendikten sonra

Değişiklikleri yollamadan önce Git işlemlerini bir araya toplar.

Git işlemleri, değişiklikler yollanmadan önce aşağıdaki kiplerden biri kullanılarak bir araya toplanabilir:

- Tüm işlemler bir arada
- Dile göre
- Dosyaya göre
- Katkıda bulunana göre

Özgün işleme iletileri korunur. Ancak *Katkıda bulunana göre* olarak seçilmediği ya da işleme iletisi eklenecek şekilde özelleştirilmediği sürece katkıda bulunan bilgisi eksik kalır.

Özgün işleme iletileri, isteğe bağlı olarak özel bir işleme iletisiyle değiştirilebilir.

Art bilgileri (Co-authored-by: ... gibi işleme satırları) isteğe bağlı olarak özgün işleme iletilerinden kaldırılabilir ve bir araya toplanmış işleme iletisinin sonuna eklenebilir. Bu aynı zamanda her çevirmen için özel bir “Co-authored-by:” bilgisi oluşturur.

JSON çıktısını özelleştirme

Eklenti kimliği

weblate.json.customize

Yapılandırma

sort_key	JSON anahtarlarını sıralama	
indent	JSON girintisi	
style	JSON girintisinin biçimi	Kullanılabilecek seçenekler: spaces – Boşluklar tabs – Sekmeler

Tetikleyiciler

depolamaya yüklendikten sonra

JSON çıktısındaki girinti ya da sıralama gibi davranışların ayarlanmasını sağlar.

Java özellikleri dosyasını biçimlendirme

Eklenti kimliği

`weblate.properties.sort`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depoya işlenmeden önce

Java özellikleri dosyasını biçimlendirir ve sıralar.

- Yeni satır karakterlerini Unix karakterleri olarak değiştirir.
- Unicode kaçış dizilerini büyük harfle biçimlendirir (varsa).
- Boş satırları ve yorumları siler.
- Dizgeleri anahtarlara göre sıralar.
- Yinelenen dizgeleri kaldırır.

Eski yorumları kaldırma

3.7 sürümünde geldi.

Eklenti kimliği

`weblate.removal.comments`

Yapılandırma

age	Tutulacak gün sayısı
-----	----------------------

Tetikleyiciler

günlük

Yorumların kaldırılması için bir zaman dilimi ayarlayabilirsiniz.

Bu özellik, güncelliğini yitirmiş olabilecek eski yorumları kaldırmak için yararlı olabilir. Bir yorumun eski olması önemini yitirdiği anlamına gelmediğinden dikkatli kullanın.

Eski önerileri kaldırma

3.7 sürümünde geldi.

Eklenti kimliği

`weblate.removal.suggestions`

Yapılandırma

age	Tutulacak gün sayısı	
votes	Oy verme eşiği	Kaldırma eşiği. Oylama kapalıyken bu alanın herhangi bir etkisi yoktur.

Tetikleyiciler

günlük

Önerilerin kaldırılması için bir zaman dilimi ayarlayabilirsiniz.

Belirli bir zaman diliminde yeterince olumlu oy almayan önerilerin kaldırılması için öneri oylamayla (bilgi almak için [Akran değerlendirmesi](#)) bağlantılı olarak çok yararlı olabilir.

RESX dosyalarını güncelleme

3.9 sürümünde geldi.

Eklenti kimliği

`weblate.resx.update`

Yapılandırma

Bu eklentinin yapılandırması yok.

Tetikleyiciler

depo güncellenmesinden sonra

Tüm çeviri dosyalarını tek dilli temel dosya ile eşleştirecek biçimde günceller. Kullanılmayan dizgeler kaldırılır ve yeni dizgeler kaynak dizgenin kopyaları olarak eklenir.

İpucu: Yalnızca eski çeviri anahtarlarını kaldırmak istiyorsanız [Çeviri dosyalarını temizleme](#) kullanın.

Ayrıca bakınız:

Does Weblate update translation files besides translations?

XML çıktısını özelleştirme

4.15 sürümünde geldi.

Eklenti kimliği

`weblate.xml.customize`

Yapılandırma

<code>closing_tags</code>	Boş XML kod imlerine kapanış kod imi eklensin
---------------------------	---

Tetikleyiciler

depolamaya yüklendikten sonra

YAML çıktısında satır uzunluğu ve yeni satır gibi davranışların ayarlanmasını sağlar.

YAML çıktısını özelleştirme

3.10.2 sürümünde geldi.

Eklenti kimliği

`weblate.yaml.customize`

Yapılandırma

in-	YAML	
dent	girintisi	
width	Uzun	Kullanılabilir seçenekler:
	satırları	80 – Satırlar 80 karakterde kaydırılsın
	kaydır-	100 – Satırları 100 karakterde kaydırılsın
	ma	120 – Satırlar 120 karakterde kaydırılsın
		180 – Satırlar 180 karakterde kaydırılsın
		65535 – Satırlar kaydırılmasın
line_	Satır	Kullanılabilir seçenekler:
end:	sonları	dos – DOS (\r\n)
		unix – UNIX (\n)
		mac – MAC (\r)

Tetikleyiciler

depolamaya yüklendikten sonra

YAML çıktısında satır uzunluğu ve yeni satır gibi davranışların ayarlanmasını sağlar.

2.14.2 Eklenti listesini özelleştirme

Eklenti listesi `WEBLATE_ADDONS` tarafından yapılandırılır. Başka bir eklenti eklemek için, bu seçeneğe mutlak sınıf adını eklemeniz yeterlidir.

2.14.3 Eklenti yazma

Kendi eklentilerinizi de yazabilir, eklenti üst verilerini tanımlamak için `weblate.addons.base.BaseAddon` alt sınıfını oluşturabilir ve ardından işlemi yapmak için bir geri çağırma ekleyebilirsiniz.

Ayrıca bakınız:

Eklentileri geliştirmek

2.14.4 Eklentiden betikleri çalıştırma

Eklentiler, dış betikleri yürütmek için de kullanılabilir. Bu özellik eskiden Weblate ile bütünleşikti. Ancak şimdi betikleri bir eklenti ile sarmalamak için bazı kodlar yazmanız gerekiyor.

```
# 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
    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"
```

Kurulum hakkında bilgi almak için *Özel kalite denetimleri, eklentiler ve kendiliğinden düzeltmeler*.

Betik dosyası, belirtilen bileşen için sürüm denetimi sistemi deposunun kök klasörüne ayarlanmış geçerli klasör ile yürütülür.

Ek olarak, şu ortam değişkenleri kullanılabilir:

WL_VCS

Kullanılan sürüm denetimi sistemi.

WL_REPO

Yukarı akış deposu adresi.

WL_PATH

Sürüm denetimi sistemi deposunun mutlak yolu.

WL_BRANCH

2.11 sürümünde geldi.

Geçerli bileşende yapılandırılmış depo dalı.

WL_FILEMASK

Geçerli bileşen için dosya maskesi.

WL_TEMPLATE

Tek dilli çeviriler için kalıbın dosya adı (boş olabilir).

WL_NEW_BASE

2.14 sürümünde geldi.

Yeni çeviriler oluşturmak için kullanılan dosyanın adı (boş olabilir).

WL_FILE_FORMAT

Geçerli bileşende kullanılan dosya biçimi.

WL_LANGUAGE

İşlenmekte olan çevirinin dili (bileşen düzeyinde kancalar için kullanılamaz).

WL_PREVIOUS_HEAD

Güncellemeden sonra önceki HEAD (yalnızca güncelleme sonrası kancası çalıştırıldıktan sonra kullanılabilir).

WL_COMPONENT_SLUG

3.9 sürümünde geldi.

Adresi oluşturmak için kullanılan bileşen kısaltması.

WL_PROJECT_SLUG

3.9 sürümünde geldi.

Adresi oluşturmak için kullanılan proje kısaltması.

WL_COMPONENT_NAME

3.9 sürümünde geldi.

Bileşen adı.

WL_PROJECT_NAME

3.9 sürümünde geldi.

Proje adı.

WL_COMPONENT_URL

3.9 sürümünde geldi.

Bileşen adresi.

WL_ENGAGE_URL

3.9 sürümünde geldi.

Proje etkileşim adresi.

Ayrıca bakınız:

Bileşen yapılandırması

Güncelleme sonrası depo işleme

Sürüm denetimi sistemi yukarı akış kaynağı değiştiğinde çeviri dosyalarını güncellemek için kullanılabilir. Bunu yapabilmek için lütfen Weblate tarafından yalnızca sürüm denetimi sistemine işlenmiş dosyaların görüldüğünü, bu nedenle değişiklikleri betiğin bir parçası olarak yapmanız gerektiğini unutmayın.

Örneğin Gulp ile bunu şu kodu kullanarak yapabilirsiniz:

```
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Depoya işlenmeden önce çevirilerin işlenmesi

Bir çevirinin depoya kaydedilmesinden önce kendiliğinden değiştirilmesi için işleme betiğini kullanın.

Geçerli bir çevirinin dosya adından oluşan tek bir parametre olarak aktarılır.

2.15 Çeviri belleği

2.20 sürümünde geldi.

Weblate comes with a built-in translation memory consisting of the following:

- Manually imported translation memory (see *Kullanıcı arayüzü*).
- Automatically stored translations performed in Weblate (depending on *Çeviri belleği kapsamı*).
- Automatically imported past translations.

Content in the translation memory can be applied one of two ways:

- El ile, çeviri yapılırken *Kendiliğinden öneriler* görünümü ile.
- Kendiliğinden, dizgeleri *Kendiliğinden çeviri* ya da *Kendiliğinden çeviri* eklentisi ile çevirerek.

For installation tips, see *Weblate Çeviri Belleği*, which is turned on by default.

2.15.1 Çeviri belleği kapsamı

3.2 sürümünde geldi: 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.

İçe aktarılan çeviri belleği

Importing arbitrary translation memory data using the *import_memory* command makes memory content available to all users and projects.

Kullanıcıların bireysel çeviri belleği

Stores all user translations automatically in the personal translation memory of each respective user.

Projelerin bireysel çeviri belleği

All translations within a project are automatically stored in a project translation memory only available for this project.

Paylaşılan çeviri belleği

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 Çeviri belleğini yönetmek

Kullanıcı arayüzü

3.2 sürümünde geldi.

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.

İpucu: Translation memory in JSON can be imported into Weblate, TMX is provided for interoperability with other tools.

Ayrıca bakınız:

Weblate çeviri belleği şeması

The screenshot shows the Weblate user interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the user 'testuser' is logged in, and the 'Translation memory' section is active. The 'Translation memory status' section shows 'Number of your entries' as 0 and 'Total number of entries' as 0. There are buttons for 'Download as JSON', 'Download as TMX', and 'Delete'. Below this is the 'Import translation memory' section, which has a 'File' input field with a 'Choose File' button and the text 'No file chosen'. A note states 'You can upload a TMX or JSON file.' and there is an 'Upload' button.

Yönetim arayüzü

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 Yapılandırma

Tüm ayarlar `settings.py` dosyasında (Django için her zamanki gibi) tutulur.

Not: Bu ayarlardan herhangi birini değiştirdikten sonra, Weblate WSGI ve Celery işlemlerini yeniden başlatmanız gerekir.

`mod_wsgi` olarak çalıştırılması durumunda, yapılandırmayı yeniden yüklemek için Apache sunucusunu yeniden başlatmanız gerekir.

Ayrıca bakınız:

Django yapılandırma parametreleri için [Django belgelerine](#) bakabilirsiniz.

2.16.1 AKISMET_API_KEY

Weblate, istenmeyen ileti olarak gelen anonim önerileri denetlemek için Akismet kullanabilir. Bir API anahtarı satın almak ve bunu bir siteyle ilişkilendirmek için ‘akismet.com <<https://akismet.com/>>’ adresine bakabilirsiniz.

2.16.2 ANONYMOUS_USER_NAME

Oturum açmamış kullanıcılar için kullanıcı adı.

Ayrıca bakınız:

Erişim denetimi

2.16.3 AUDITLOG_EXPIRY

3.6 sürümünde geldi.

Weblate hesap etkinliği hakkında bilgi içeren denetim günlüklerinin tutulacağı gün sayısı.

Varsayılan değer: 180 gün.

2.16.4 AUTH_LOCK_ATTEMPTS

2.14 sürümünde geldi.

Hız sınırlaması uygulanmadan önce başarısız olmasına izin verilecek en fazla kimlik doğrulama girişimi sayısı.

Bu seçenek şu anda şu konumlarda uygulanmaktadır:

- Oturum aç. Hesap parolasını silerek kullanıcının yeni bir parola isteği yapmadan oturum açmasını engeller.
- Parola sıfırlama. Çok fazla parola sıfırlama girişimi ile istenmeyen trafik oluşturan kullanıcılara yeni e-posta gönderilmesini engeller.

Varsayılan değer: 10.

Ayrıca bakınız:

Hız sınırlama

2.16.5 AUTO_UPDATE

3.2 sürümünde geldi.

3.11 sürümünde değişti: Özgün açma/kapama seçeneği, hangi dizgelerin kabul edildiğini ayırt edecek biçimde değiştirildi.

Tüm depoları günlük olarak günceller.

İpucu: Weblate depolarını kendiliğinden güncellemek için *Bildirim kancaları* kullanmıyorsanız yararlıdır.

Not: Geriye dönük uyumluluk için açma/kapama seçenekleri dizge seçimine ek olarak bulunur.

Seçenekler:

"none"

Günlük güncelleme yapılmaz.

"remote" ayrıca False

Yalnızca uzak tarafı günceller.

"full" ayrıca True

Uzak tarafı günceller ve çalışan kopya ile birleştirir.

Not: Bunun için *Celery ile arka plan görevlerini kullanmak* çalışıyor olmalıdır. Yeniden başlatıldıktan sonra etkili olur.

2.16.6 AVATAR_URL_PREFIX

Avatar adreslerini şu biçimde oluşturmak için ön ek: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Şu hizmetlerin çalıştığı biliniyor:

Gravatar (varsayılan), <https://gravatar.com/> üzerinden

`AVATAR_URL_PREFIX = 'https://www.gravatar.com/'`

Libravatar, <https://www.libravatar.org/> üzerinden

`AVATAR_URL_PREFIX = 'https://www.libravatar.org/'`

Ayrıca bakınız:

Avatar ön belleği, ENABLE_AVATARS, Avatarlar

2.16.7 AUTH_TOKEN_VALID

2.14 sürümünde geldi.

Kimlik doğrulama belirtecinin ve parola sıfırlama e-postalarındaki geçici parolanın geçerli olacağı süre. Varsayılan olarak 172800 saniye (2 gün).

2.16.8 AUTH_PASSWORD_DAYS

2.15 sürümünde geldi.

Bir kullanıcının daha önce kullandığı bir parolayı yeniden kullanabilmesi için geçmesi gereken gün sayısı.

Bu işlem denetim günlüğü ile yapılır. `AUDITLOG_EXPIRY` seçeneğinin değeri en az bu değer ile aynı olmalıdır.

Not: Weblate 2.15 sürümünden önce yapılan parola değişiklikleri bu ilkede dikkate alınmaz.

Varsayılan değer: 180 gün.

2.16.9 AUTOFIX_LIST

Bir çeviri kaydedilirken uygulanacak kendiliğinden düzeltmelerin listesi.

Not: Kendiliğinden düzeltme arayüzünü uygulayan Python sınıfının tam yolunu belirtin.

Kullanılabilecek düzeltmeler:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace`

Çevirinin başındaki ve sonundaki boşlukları kaynak ile eşleştirir.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis`

Kaynakta üç nokta karakteri (...) varsa çevirinin sonundaki üç ayrı noktayı (...) değiştirir.

`weblate.trans.autofixes.chars.RemoveZeroSpace`

Kaynakta bulunmuyorsa çevirideki sıfır genişlikli boşluk karakterlerini kaldırır.

`weblate.trans.autofixes.chars.RemoveControlChars`

Kaynakta bulunmuyorsa denetim karakterlerini kaldırır.

`weblate.trans.autofixes.html.BleachHTML`

Güvenli olmayan HTML kod imlerini `safe-html` olarak işaretlenmiş çevirilerden kaldırır (bilgi almak için: *Güvenli olmayan HTML*).

Hangilerinin kullanılacağını seçebilirsiniz:

```
AUTOFIX_LIST = (  
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",  
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",  
)
```

Ayrıca bakınız:

Automatic fixups, Özel kendiliğinden düzeltmeler

2.16.10 BACKGROUND_TASKS

4.5.2 sürümünde geldi.

Bir bileşen için uzun bakım görevlerinin tetiklenme sıklığını belirtir.

Şu anda bunun denetlediği:

- *Kendiliğinden çeviri* eklentisi
- *Denetimler ve düzeltmeler* yeniden hesaplama

Olası seçenekler:

- `aylık` (varsayılan)
- `weekly`
- `daily`
- `never`

Not: Weblate üzerinde binlerce bileşen bulunuyorsa sıklığın artırılması önerilmez.

2.16.11 BASIC_LANGUAGES

4.4 sürümünde geldi.

Kullanıcılara yeni çeviriye başlamaları için sunulacak dillerin listesi. Belirtilmediğinde, yaygın olarak kullanılan tüm dilleri içeren, ancak ülkeye özgü lehçeleri olmayan iç liste kullanılır.

Bu seçenek, yalnızca yetki verilmemiş kullanıcıların istenmeyen diller eklemesini engeller. Proje yöneticilerine Weblate üzerinde tanımlanmış dillerin tümü sunulur.

Not: Bu seçenek, Weblate üzerinde yeni diller tanımlamaz, yalnızca veri tabanında var olan dilleri süzer.

Örnek:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Ayrıca bakınız:

Dil tanımları

2.16.12 BORG_EXTRA_ARGS

4.9 sürümünde geldi.

İç yedeklemeler tetiklendiğinde **borg create** komutuna ek parametreler iletebilirsiniz.

Örnek:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Ayrıca bakınız:

Weblate yedeğini alma ve taşıma, `borg create`

2.16.13 CACHE_DIR

4.16 sürümünde geldi.

Weblate ön bellek dosyalarının tutulacağı klasör. Varsayılan olarak `DATA_DIR` içindeki `cache` alt klasörü kullanılır.

`DATA_DIR` bir ağ dosya sisteminde bulunuyorsa bunu yerel ya da geçici bir dosya sistemi olarak değiştirin.

Docker kapsayıcısı bunun için ayrı bir birim kullanır. Bilgi almak için: `:ref: docker-volume`.

2.16.14 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Weblate için Content-Security-Policy üst bilgisini özelleştirin. Başlık, üçüncü taraf hizmetlerle (Matomo, Google Analytics, Sentry, ...) etkin bütünleştirmelere göre kendiliğinden oluşturulur.

Tüm bunlar varsayılan olarak boş listedir.

Örnek:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Ayrıca bakınız:

İçerik güvenliği ilkesi, İçerik güvenliği ilkesi (CSP)

2.16.15 CHECK_LIST

Bir çeviride yapılacak hata denetimlerinin listesi.

Not: Denetim arayüzünü uygulayan Python sınıfının tam yolunu yazın.

Denetim listesini, isteğinize göre ayarlayın.

Tüm iç *Quality checks* varsayılan olarak açıktır ve buradan bu ayarları değiştirebilirsiniz. Varsayılan olarak, kullanılabilecek değerler *Örnek yapılandırma* dosyası içinde yorum satırı olarak bulunurlar. Daha sonra her yeni Weblate sürümü için yeni denetimler yapıldı.

Tüm denetimleri kapatabilirsiniz:

```
CHECK_LIST = ()
```

Yalnızca birkaçını açabilirsiniz:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Not: Bu ayarın değiştirilmesi yalnızca yeni değiştirilen çevirileri etkiler. Var olan denetimlerin veri tabanında tutulması sürdürülür. Değişiklikleri kaydedilmiş çevirilere de uygulamak için *updatechecks* komutunu yürütün.

Ayrıca bakınız:

Quality checks, Parametreleri kullanarak davranışı özelleştirmek

2.16.16 COMMENT_CLEANUP_DAYS

3.6 sürümünde geldi.

Belirli günden sonra yorumları siler. Varsayılan değer `Yok`. Yani bir silme işlemi yapılmaz.

2.16.17 COMMIT_PENDING_HOURS

2.10 sürümünde geldi.

Bekleyen değişikliklerin gönderilmesi ile arka plan görevi tarafından işlenmesi arasındaki saat sayısı.

Ayrıca bakınız:

Bileşen yapılandırması, İşlenecek değişikliklerin yaşı, Bakım görevlerini yürütmek, `commit_pending`

2.16.18 CONTACT_FORM

4.6 sürümünde geldi.

İletişim formu e-postalarının nasıl gönderileceğini yapılandırır. E-posta sunucusu yapılandırmanıza uygun bir yapılandırma seçin.

"reply-to"

Gönderici *Yanıt* olarak kullanılır. Varsayılan davranış budur.

"from"

Gönderici *Kimden* olarak kullanılır. E-posta sunucunuz bu tür e-postaların gönderilmesine izin vermelidir.

2.16.19 DATA_DIR

Weblate klasöründe tüm veriler bulunur. Sürüm denetimi sistemi depo bağlantılarını, tam metin dizinini ve dış araçlar için çeşitli yapılandırma dosyalarını içerir.

Genellikle aşağıdaki alt klasörler de bulunur:

home

Betikleri çağırmak için kullanılan giriş klasörü.

ssh

SSH anahtarları ve yapılandırması.

static

Durağan Django dosyalarının varsayılan konumu `STATIC_ROOT` tarafından belirlenir. Bilgi almak için: `:ref:`static-files``.

Docker kapsayıcısı bunun için ayrı bir birim kullanır. Bilgi almak için: `:ref:`docker-volume``.

media

Django ortam dosyalarının varsayılan konumu `MEDIA_ROOT` tarafından belirlenir. Yüklenen ekran görüntülerini içerir. Bilgi almak için: `:ref:`screenshots``.

vcs

Çeviriler için sürüm denetimi depoları.

backups

Günlük yedekleme verileri. Bilgi almak için: *Yedeklemeler için dökümü yapılan veriler*.

fonts:

Kullanıcı tarafından yüklenen yazı tipleri. Bilgi almak için: `:ref:`fonts``.

cache

Çeşitli ön bellekler, `CACHE_DIR` seçeneği kullanılarak başka bir konuma taşınabilir.

Docker kapsayıcısı bunun için ayrı bir birim kullanır. Bilgi almak için: `:ref:`docker-volume``.

Not: Bu klasör Weblate tarafından yazılabilir olmalıdır. uWSGI olarak çalıştırıldığında, `www-data` kullanıcısının bu klasöre yazma erişimi olmalıdır.

Bunu yapmanın en kolay yolu, kullanıcıyı klasörün sahibi yapmaktır:

```
sudo chown www-data:www-data -R $DATA_DIR
```

Varsayılan değer: `/home/weblate/data`. Ancak yapılandırılması beklenir.

Ayrıca bakınız:

Dosya sistemi izinleri, Weblate yedeğini alma ve taşıma, `CACHE_DIR`

2.16.20 DATABASE_BACKUP

3.1 sürümünde geldi.

Veri tabanı yedeklemelerinin düz metin ya da sıkıştırılmış olarak depolanacağını ya da depolanmayacağını belirler. Şu değerler kullanılabilir:

- "plain"
- "compressed"
- "none"

Ayrıca bakınız:

Weblate yedeğini alma ve taşıma

2.16.21 DEFAULT_ACCESS_CONTROL

3.3 sürümünde geldi.

Yeni projeler için varsayılan erişim denetimi ayarı:

0	<i>Herkese açık</i>
1	<i>Korunmuş</i>
100	<i>Kişisel</i>
200	<i>Özel</i>

Erişim denetimi listesini el ile yönetiyorsanız `:guilabel:`Özel`` olarak ayarlayın. Bu seçenek iç Weblate yönetimine güvenilmemesi anlamına gelir.

Ayrıca bakınız:

Proje erişim denetimi, Erişim denetimi

2.16.22 DEFAULT_AUTO_WATCH

4.5 sürümünde geldi.

Katkıda bulunulan projeler kendiliğinden izlensin seçeneğinin yeni kullanıcılar için açık olup olmayacağını belirler. Varsayılan değer: `True`.

Ayrıca bakınız:

Bildirimler

2.16.23 DEFAULT_RESTRICTED_COMPONENT

4.1 sürümünde geldi.

Varsayılan bileşen kısıtlaması değeri.

Ayrıca bakınız:

Kısıtlı erişim, Takımların kapsamı

2.16.24 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DEFAULT_MERGE_MESSAGE

Farklı işlemler için varsayılan işleme iletileri. Bilgi almak için: ref: *component*.

Ayrıca bakınız:

Kalıp kodlaması, Bileşen yapılandırması, İşleme, ekleme, silme, birleştirme, eklenti ve birleştirme istek iletileri

2.16.25 DEFAULT_ADDONS

Oluşturulan her bileşene varsayılan olarak kurulacak eklentiler.

Not: Bu ayar yalnızca yeni oluşturulan bileşenleri etkiler.

Örnek:

```
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",
    },
}
```

Ayrıca bakınız:

install_addon, Eklentiler, WEBLATE_ADDONS

2.16.26 DEFAULT_COMMITER_EMAIL

2.4 sürümünde geldi.

Varsayılan işleyenin e-posta adresi: `noreply@weblate.org`.

Ayrıca bakınız:

DEFAULT_COMMITER_NAME

2.16.27 DEFAULT_COMMITER_NAME

2.4 sürümünde geldi.

Varsayılan işleyenin adı: `Weblate`.

Ayrıca bakınız:

DEFAULT_COMMITER_EMAIL

2.16.28 DEFAULT_LANGUAGE

4.3.2 sürümünde geldi.

Kaynak dil içinde örnek olarak kullanılacak varsayılan kaynak dil.

Varsayılan değer: *en*. Eşleşen dil nesnesinin veri tabanında bulunması gerekir.

Ayrıca bakınız:

Dil tanımları, Kaynak dil

2.16.29 DEFAULT_MERGE_STYLE

3.4 sürümünde geldi.

Tüm yeni bileşenler için birleştirme biçemi.

- *rebase* - default
- *merge*

Ayrıca bakınız:

Bileşen yapılandırması, Birleştirme biçemi

2.16.30 DEFAULT_SHARED_TM

3.2 sürümünde geldi.

Paylaşılan çeviri belleği kullanılsın ve *Paylaşılan çeviri belleğine katkıda bulunulsun* varsayılan değerlerini yapılandırır.

2.16.31 DEFAULT_TRANSLATION_PROPAGATION

2.5 sürümünde geldi.

Varsayılan çeviri yayılması ayarı. Varsayılan değer `True`.

Ayrıca bakınız:

Bileşen yapılandırması, Çevirilerin yayılmasını sağlamak

2.16.32 DEFAULT_PULL_MESSAGE

Çekme isteklerinin varsayılan başlığını ve iletisini yapılandırır.

2.16.33 ENABLE_AVATARS

Kullanıcıların Gravatar avatarlarının kullanılıp kullanılmayacağını belirler. Varsayılan olarak kullanılabilir.

Avatarlar sunucu tarafından alınır ve ön belleğe eklenir. Böylece kişisel bilgilerin sızma riski azaltılır ve kullanıcı deneyimi hızlandırılır.

Ayrıca bakınız:

Avatar ön belleği, AVATAR_URL_PREFIX, Avatarlar

2.16.34 ENABLE_HOOKS

Anonim uzak kancaların etkinleştirilmesini sağlar.

Ayrıca bakınız:

Bildirim kancaları

2.16.35 ENABLE_HTTPS

HTTPS ya da HTTP olarak kullanılacak Weblate bağlantısı iletişim kuralı. Bu seçenek, gönderilen e-postaları ve oluşturulan mutlak adresleri etkiler.

Varsayılan yapılandırmada bu seçenek, HTTPS ile ilgili çeşitli Django ayarları için de kullanılır. Güvenli çerezleri etkinleştirir, HSTS seçeneğini açar ya da HTTPS adresine yönlendirmeyi etkinleştirir.

HTTPS yönlendirmesi bazı durumlarda sorunlu olabilir ve iletişim kuralı üst bilgilerini Django üzerine doğru şekilde iletmeyen SSL sonlandırma işlemi yapan ters bir vekil sunucu kullanmanız durumunda, sonsuz yönlendirme sorunu yaşayabilirsiniz. Lütfen ters vekil sunucu yapılandırmanızı `X-Forwarded-Proto` veya ``Forwarded`` üst bilgilerini yayacak şekilde ayarlayın ya da SSL durumunun Django tarafından doğru bir şekilde algılanmasını sağlamak için `SECURE_PROXY_SSL_HEADER` seçeneğini yapılandırın.

Ayrıca bakınız:

`SESSION_COOKIE_SECURE`, `CSRF_COOKIE_SECURE`, `SECURE_SSL_REDIRECT`, `SECURE_PROXY_SSL_HEADER` *Doğru site etki alanını ayarlayın*

2.16.36 ENABLE_SHARING

Kullanıcıların çeviri ilerlemesini sosyal ağlarda paylaşabilmesi için *Paylaş* menüsünü açar ya da kapatır.

2.16.37 EXTRA_HTML_HEAD

4.15 sürümünde geldi.

HTML üst bilgisine ek biçimlendirme ekler. Sitenin sahipliğini doğrulamak için kullanılabilir. Örneğin:

```
EXTRA_HTML_HEAD = '<link href ="https://fosstodon.org/@weblate" rel ="me">'
```

Uyarı: Dizge üzerinde herhangi bir temizleme işlemi yapılmaz. HTML üst bilgisine olduğu gibi eklenir.

2.16.38 GET_HELP_URL

4.5.2 sürümünde geldi.

Weblate kopyanızın destek bilgilerinin bulunduğu adres.

2.16.39 GITEA_CREDENTIALS

4.12 sürümünde geldi.

Gitea sunucuları için kimlik doğrulama bilgileri listesi.

```
GITEA_CREDENTIALS = {
  "try.gitea.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitea.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Ayrıca bakınız:

Gitea çekme isteği, Gitea kişisel erişim kodu oluşturmak

2.16.40 GITLAB_CREDENTIALS

4.3 sürümünde geldi.

GitLab sunucuları için kimlik doğrulama bilgileri listesi.

```
GITLAB_CREDENTIALS = {
  "gitlab.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitlab.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
} ,  
}
```

Ayrıca bakınız:

GitLab birleştirme istekleri, GitLab: Kişisel erişim kodu

2.16.41 GITHUB_CREDENTIALS

4.3 sürümünde geldi.

GitHub sunucuları için kimlik doğrulama bilgileri listesi.

```
GITHUB_CREDENTIALS = {  
  "api.github.com": {  
    "username": "weblate",  
    "token": "your-api-token",  
  },  
  "github.example.com": {  
    "username": "weblate",  
    "token": "another-api-token",  
  },  
}
```

Ayrıca bakınız:

GitHub çekme istekleri, GitHub kişisel erişim kodu oluşturmak

2.16.42 BITBUCKETSERVER_CREDENTIALS

4.16 sürümünde geldi.

Bitbucket sunucuları için kimlik doğrulama bilgileri listesi.

```
BITBUCKETSERVER_CREDENTIALS = {  
  "git.self-hosted.com": {  
    "username": "weblate",  
    "token": "http-access-token",  
  },  
}
```

Ayrıca bakınız:

Bitbucket sunucusu çekme isteği, Bitbucket: HTTP erişim kodu

2.16.43 GOOGLE_ANALYTICS_ID

Google Analytics ile Weblate izlemesini açmak için Google Analytics kimliği.

2.16.44 HIDE_REPO_CREDENTIALS

Depo kimlik bilgilerini web arayüzünde gizler. İçinde kullanıcı adı ve parolası geçen bir depo adresiniz varsa, Weblate ilgili bilgileri kullanıcılara görüntülerken bunları gizler.

Örneğin, ``https://kullanici:parola@git.ornek.com/depo.git`` yerine yalnızca ``https://git.ornek.com/depo.git`` görüntülenir. Sürüm denetimi sistemi hata iletilerinde de benzer ayıklama yapılmaya çalışılır.

Not: Bu seçenek varsayılan olarak kapalıdır.

2.16.45 HIDE_VERSION

4.3.1 sürümünde geldi.

Sürüm bilgilerini kimliği doğrulanmamış kullanıcılardan gizler. Bu seçenek aynı zamanda tüm belge bağlantılarının kurulu sürüm belgeleri yerine son sürümün belgelerinin göstermesini sağlar.

Sürümün gizlenmesi bazı kuruluşlar için önerilen bir güvenlik önlemidir. Ancak saldırganın uygulamanın davranışını inceleyerek sürümü öğrenmesini engellemez.

Not: Bu seçenek varsayılan olarak kapalıdır.

2.16.46 INTERLEDGER_PAYMENT_POINTERS

4.12.1 sürümünde geldi.

Web üzerinden para kazanmak için genel muhasebe ödeme göstergeleri (ILP) listesi.

Birden fazla belirtilirse, rastgele biri seçilerek olasılıksal gelir paylaşımı sağlanır.

Ayrıntılı bilgi almak için <<https://webmonetization.org/>> sitesine bakabilirsiniz.

İpucu: Varsayılan değer, kullanıcıların Weblate için fon sağlamasına yardımcı olur.

2.16.47 IP_BEHIND_REVERSE_PROXY

2.14 sürümünde geldi.

Weblate platformunun ters vekil sunucu arkasında çalışıp çalışmadığını gösterir.

True olarak ayarlanırsa, Weblate IP adresini `IP_PROXY_HEADER` seçeneğindeki bir üst bilgidен alır.

Uyarı: Gerçekten bir ters vekil sunucu kullandığınızdan ve bu üst bilginin ayarlandığından emin olun. Yoksa kullanıcılar IP adresini taklit edebilir.

Not: Bu seçenek, varsayılan olarak kapalıdır.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, Hız sınırlama, IP_PROXY_HEADER, IP_PROXY_OFFSET

2.16.48 IP_PROXY_HEADER

2.14 sürümünde geldi.

`IP_BEHIND_REVERSE_PROXY` seçeneği açıldığında hangi Weblate üst bilgisinde IP adresinin bulunduğunu gösterir.

Varsayılan değer: `HTTP_X_FORWARDED_FOR`.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, Hız sınırlama, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_OFFSET

2.16.49 IP_PROXY_OFFSET

2.14 sürümünde geldi.

:setting:'IP_PROXY_HEADER' ögesinin istemci IP adresi olarak kullanılacak bölümünü gösterir.

Kurulumunuza bağlı olarak, bu üst bilgi birkaç IP adresinden oluşabilir (örneğin, X-Forwarded-For: a, b, client-ip) ve üst bilgidен hangi adresin istemci IP adresi olarak kullanılacağını bu bölümden yapılandırabilirsiniz.

Uyarı: Bunu ayarlamak kurulumunuzun güvenliğini etkiler. Yalnızca IP adresini belirlemek için güvenilen vekil sunucuları kullanacak şekilde yapılandırmanız gerekir.

Varsayılan değer: 0.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, Hız sınırlama, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER

2.16.50 LEGAL_TOS_DATE

4.15 sürümünde geldi.

Not: Bunun çalışması için *Yasal* kurulmuş olmalıdır.

Hizmet koşulları belgelerinin son güncellenme tarihi. Tarih değiştiğinde, kullanıcıların hizmet koşullarını kabul etmeleri gerekir.

```
from datetime import date

LEGAL_TOS_DATE = date(2022, 2, 2)
```

2.16.51 LEGAL_URL

3.5 sürümünde geldi.

Weblate kopyanızın yasal belgelerinin bulunduğu adres.

İpucu: Yasal belgelerinizi Weblate dışında bulunduruyor ve Weblate içine eklemek istiyorsanız yararlıdır. Bilgi almak için: *Yasal*.

Örnek:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Ayrıca bakınız:

`PRIVACY_URL`

2.16.52 LICENSE_EXTRA

Lisans seçeneklerine katılacak ek lisanslar.

Not: Her lisans tanımında bir kısa ad, bir uzun ad ve bir adres olmalıdır.

Örnek:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

2.16.53 LICENSE_FILTER

4.3 sürümünde değişti: Bu seçenek boş olarak ayarlandığında lisans uyarısını kapatır.

Görüntülenecek lisans listesini süzer. Bu seçenek, boş olarak ayarlandığında lisans uyarısını da kapatır.

Not: Bu süzgeç kısa lisans adlarını kullanır.

Örnek:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

Şu lisans uyarısını kapatır:

```
LICENSE_FILTER = set()
```

Ayrıca bakınız:

`alerts`

2.16.54 LICENSE_REQUIRED

Bileşen yapılandırması içindeki lisans özniteliğinin zorunlu olup olmadığını belirler.

Not: Bu seçenek varsayılan olarak kapalıdır.

2.16.55 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Belirtilen bir çevirinin uzunluğunun sınırlı olmasının gerekip gerekmediği. Kısıtlama, kaynak dizgenin uzunluğu x 10 karakterdir.

İpucu: Kaynak dizge uzunluğuna bakılmaksızın daha uzun çevirilere (10.000 karaktere kadar) izin vermek için bu seçeneği `False` olarak ayarlayın.

Not: Varsayılan değer: `True`.

2.16.56 LOCALIZE_CDN_URL ve LOCALIZE_CDN_PATH

Bu ayarlar *JavaScript yerelleştirme CDN* eklentisini yapılandırır. `LOCALIZE_CDN_URL`, yerelleştirme CDN sunucusunun kullanılabilir kök adresini belirler ve `LOCALIZE_CDN_PATH` seçeneği, `LOCALIZE_CDN_URL` adresinde sunulan oluşturulmuş dosyaların kaydedileceği yolu belirler.

İpucu: Hosted Weblate üzerinde bu seçenek, `https://weblate-cdn.com/` adresini kullanır.

Ayrıca bakınız:

JavaScript yerelleştirme CDN

2.16.57 LOGIN_REQUIRED_URLS

Oturum açmanın dayatılmasını istediğiniz adreslerin listesi. (Weblate üzerindeki iç standart kuralların yanı sıra).

İpucu: Bu seçenek, şunu kullanarak tüm kurulumu parola ile korumanızı sağlar:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

İpucu: Yukarıdaki örnekte gösterildiği gibi API erişiminin de kilitlenmesi istenir.

Ayrıca bakınız:

`REQUIRE_LOGIN`

2.16.58 LOGIN_REQUIRED_URLS_EXCEPTIONS

`LOGIN_REQUIRED_URLS` için istisna listesi. Belirtilmediğinde kullanıcıların oturum açma sayfasına erişmesine izin verilir.

Ekleme isteyebileceğiniz bazı istisnalar:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$", # Required for sign in
    r"/static/(.*)$", # Required for development mode
    r"/widgets/(.*)$", # Allowing public access to widgets
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
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

İzlemek istediğiniz Matomo (eski adıyla Piwik) üzerindeki bir site kimliği.

Not: Bu bütünleştirme Matomo Etiket Yöneticisi özelliğini desteklemez.

Ayrıca bakınız:

MATOMO_URL

2.16.60 MATOMO_URL

Weblate kullanımını izlemek için kullanmak istediğiniz bir Matomo (eski adıyla Piwik) kurulumunun tam adresi (sondaki bölü karakteri ile birlikte). Bilgi almak için: <<https://matomo.org/>>.

İpucu: Bu bütünleştirme Matomo Etiket Yöneticisi özelliğini desteklemez.

Örnek:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Ayrıca bakınız:

MATOMO_SITE_ID

2.16.61 NEARBY_MESSAGES

Çevrilen dizgenin yakınında görüntülenecek dizge sayısı. Bu yalnızca varsayılan bir değerdir. Kullanıcılar bu değeri *Kullanıcı profili* bölümünden ayarlayabilir.

2.16.62 DEFAULT_PAGE_LIMIT

4.7 sürümünde geldi.

Sayfalama etkin olduğunda bir sayfada varsayılan olarak görüntülenecek öge sayısı.

2.16.63 PAGURE_CREDENTIALS

4.3.2 sürümünde geldi.

Pagure sunucuları için kimlik doğrulama bilgilerinin listesi.

```
PAGURE_CREDENTIALS = {
  "pagure.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "pagure.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Ayrıca bakınız:

Pagure birleştirme istekleri, *Pagure API*

2.16.64 PRIVACY_URL

4.8.1 sürümünde geldi.

Weblate kopyanızın gizlilik ilkesinin bulunduğu adres.

İpucu: Yasal belgelerinizi Weblate dışında bulunduruyor ve Weblate içine eklemek istiyorsanız yararlıdır. Bilgi almak için: *Yasal*.

Örnek:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Ayrıca bakınız:

LEGAL_URL

2.16.65 PRIVATE_COMMIT_EMAIL_OPT_IN

4.15 sürümünde geldi.

Özel işleme e-postasının opt-in mi opt-out mu olduğunu yapılandırır (varsayılan değer: opt-out).

Ayrıca bakınız:

Profil, *PRIVATE_COMMIT_EMAIL_TEMPLATE*

2.16.66 PRIVATE_COMMIT_EMAIL_TEMPLATE

4.15 sürümünde geldi.

Bir kullanıcının özel işleme e-postasını oluşturan kalıp. Varsayılan değer: "{username}@users.noreply.{site_domain}".

Devre dışı bırakılması için boş bırakın.

Not: Farklı işleme e-postası kullanmak, *PRIVATE_COMMIT_EMAIL_OPT_IN* seçeneği ile yapılandırılmadığı sürece kullanıcılar için abone olmak anlamına gelir. Kullanıcılar işleme e-postasını *Profil* içinde yapılandırabilir.

2.16.67 PROJECT_BACKUP_KEEP_COUNT

4.14 sürümünde geldi.

Her proje için sunucuda tutulacak yedek sayısını belirler. Varsayılan değer: 3.

Ayrıca bakınız:

Proje düzeyinde yedekler

2.16.68 PROJECT_BACKUP_KEEP_DAYS

4.14 sürümünde geldi.

Proje yedeklerinin sunucuda tutulacağını süreyi belirler. Varsayılan değer: 30 gün.

Ayrıca bakınız:

Proje düzeyinde yedekler

2.16.69 PROJECT_NAME_RESTRICT_RE

4.15 sürümünde geldi.

Proje adlandırmasını kısıtlayacak kurallı ifadeyi belirler. Kurallı ifadeye uyan adlar reddedilir.

Ayrıca bakınız:

Proje adı

2.16.70 PROJECT_WEB_RESTRICT_RE

4.15 sürümünde geldi.

Proje web sitelerini kısıtlayacak kurallı ifadeyi belirler. Kurallı ifadeye uyan adresler reddedilir.

Ayrıca bakınız:

Proje web sitesi

2.16.71 RATELIMIT_ATTEMPTS

3.2 sürümünde geldi.

Hız sınırlaması uygulanmadan önce yapılabilecek en fazla kimlik doğrulama girişi sayısı.

Varsayılan değer: 5.

Ayrıca bakınız:

Hız sınırlama, RATELIMIT_WINDOW, RATELIMIT_LOCKOUT

2.16.72 RATELIMIT_WINDOW

3.2 sürümünde geldi.

Hız sınırlaması uygulandıktan sonra kimlik doğrulamasının kabul edileceği süre.

Varsayılan değer: 300 saniye (5 dakika).

Ayrıca bakınız:

Hız sınırlama, *RATELIMIT_ATTEMPTS*, *RATELIMIT_LOCKOUT*

2.16.73 RATELIMIT_LOCKOUT

3.2 sürümünde geldi.

Hız sınırlaması uygulandıktan sonra kimlik doğrulamasının kilitleneceği süre.

Varsayılan değer: 600 saniye (10 dakika).

Ayrıca bakınız:

Hız sınırlama, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*

2.16.74 REGISTRATION_ALLOW_BACKENDS

4.1 sürümünde geldi.

Hesap açılmasına izin verilecek kimlik doğrulama arka uçlarının listesi. Bu seçenek yalnızca yeni kayıtları sınırlar. Kullanıcılar yapılandırılmış tüm kimlik doğrulama arka uçları ile kimlik doğrulaması yapabilir ve kimlik doğrulaması ekleyebilir.

Hesap açma arka uçlarını sınırlarken *REGISTRATION_OPEN* seçeneğinin etkin tutulması önerilir. Yoksa kullanıcılar hesap açabilir, ancak Weblate kullanıcı arayüzüne hesap açma bağlantılarını görüntüleyemez.

Örnek:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

İpucu: Arka uç adları, kimlik doğrulaması için adres içinde kullanılan adlarla eşleşir.

Ayrıca bakınız:

REGISTRATION_OPEN, *Kimlik doğrulama*

2.16.75 REGISTRATION_CAPTCHA

Yeni hesap açılışlarının CAPTCHA tarafından korunup korunmadığını gösteren `True` ya da `False` değeri. Bu ayar isteğe bağlıdır ve belirtilmezse varsayılan olarak `True` olduğu varsayılır.

Açıkça, kullanıcıların e-posta adreslerini yazdığı tüm sayfalara bir CAPTCHA eklenir:

- Yeni hesap açılışı.
- Parola kurtarma.
- Bir hesaba e-posta ekleme.
- Oturum açmamış kullanıcılar için iletişim formu.

2.16.76 REGISTRATION_EMAIL_MATCH

2.17 sürümünde geldi.

Hangi e-posta adreslerinin kaydedilebileceğini süzmenizi sağlar.

Varsayılan değer: `.*`. Herhangi bir e-posta adresi ile hesap açılmasına izin verir.

Bu seçeneği, hesap açma işlemi için tek bir e-posta etki alanıyla sınırlamak için kullanabilirsiniz:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.77 REGISTRATION_OPEN

Şu anda yeni hesap açılmasına izin verilip verilmediği. Bu ayar isteğe bağlıdır ve varsayılan `True` değerinde kalabileceği gibi `False` olarak da değiştirilebilir.

Bu ayar, e-posta adresi ya da Python Social Auth ile yapılan iç kimlik doğrulamasını etkiler (`REGISTRATION_ALLOW_BACKENDS` komutunu kullanarak belirli arka uçları beyaz listeye ekleyebilirsiniz).

Not: *LDAP kimlik doğrulaması* gibi üçüncü taraf kimlik doğrulama yöntemleri kullanılıyorsa bu seçenek yalnızca kayıt formunu gizler. Ancak yeni kullanıcılar yine de oturum ve hesap açabilir.

Ayrıca bakınız:

REGISTRATION_ALLOW_BACKENDS, REGISTRATION_EMAIL_MATCH, Kimlik doğrulama

2.16.78 REGISTRATION_REBIND

4.16 sürümünde geldi.

Var olan kullanıcılar için kimlik doğrulama hizmeti arka uçlarının yeniden bağlanmasını sağlar. Kimlik doğrulama hizmeti sağlayıcıları arasında geçiş yaparken bu seçeneği etkinleştirin.

Not: Bu seçenek varsayılan olarak, var olan hesaba başka kimlik doğrulama hizmeti arka uçlarının eklenmesini engellemek için devre dışı bırakılmıştır. Yeniden bağlama seçeneği, daha fazla üçüncü taraf kimlik doğrulama hizmeti arka ucunun kullanılmasını sağlarken hesabın ele geçirilme riskini artırır.

2.16.79 REPOSITORY_ALERT_THRESHOLD

4.0.2 sürümünde geldi.

Eski ya da çok fazla değişiklik içeren depolar için bir uyarı tetikleme eşiği. Varsayılan değer: 25.

Ayrıca bakınız:

alerts

2.16.80 REQUIRE_LOGIN

4.1 sürümünde geldi.

Bu seçenek, `LOGIN_REQUIRED_URLS` özelliğini etkinleştirir ve REST çatısını tüm API uç noktaları için kimlik doğrulaması gerekecek şekilde yapılandırır.

Not: Bu seçenek, *Örnek yapılandırma* içine eklenmiştir. Docker için `WEBLATE_REQUIRE_LOGIN` kullanın.

2.16.81 SENTRY_DSN

3.9 sürümünde geldi.

Hata raporlarını derlemek için kullanılacak Sentry DSN.

Ayrıca bakınız:

Sentry için Django bütünleşmesi

2.16.82 SESSION_COOKIE_AGE_AUTHENTICATED

4.3 sürümünde geldi.

Kimliği doğrulanmış kullanıcılar için oturum geçerlilik süresini ayarlayın. Bu seçenek, kimliği doğrulanmamış kullanıcılar için kullanılan `SESSION_COOKIE_AGE` seçeneğini tamamlar.

Ayrıca bakınız:

`SESSION_COOKIE_AGE`

2.16.83 SIMPLIFY_LANGUAGES

Varsayılan dil/ülke kombinasyonları için basit dil kodlarının kullanılmasını sağlar. Örneğin, bir “fr_FR” çevirisi “fr” dil kodunu kullanır. Varsayılan kombinasyonlar için dillerin listelenmesini basitleştirdiğinden genellikle bu davranış istenir.

Her çeşit için farklı çeviriler yapmak istiyorsanız bu seçeneği kapatın.

2.16.84 SITE_DOMAIN

Site etki alanını yapılandırır. Bu seçenek, birçok kapsamda doğru mutlak bağlantılar üretmek için gereklidir (etkinleştirme e-postaları, bildirimler ya da RSS akışlarındaki gibi).

Weblate standart olmayan bir bağlantı noktası kullanıyorsa, buraya da ekleyin.

Örnekler:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Not: Bu ayarda yalnızca etki alanı adı bulunmalıdır. İletişim kuralını yapılandırmak için (HTTPS kullanımını etkinleştirmek ve dayatmak) `ENABLE_HTTPS` seçeneğini ve adresi değiştirmek için `URL_PREFIX` seçeneğini kullanın.

İpucu: Bir Docker kapsayıcısında, site etki alanı `WEBLATE_ALLOWED_HOSTS` seçeneği ile yapılandırılır.

Ayrıca bakınız:

Doğru site etki alanını ayarlayın, Allowed hosts kurulumu, HTTPS ayarını düzgün biçimde yapın `WEBLATE_SITE_DOMAIN`, `ENABLE_HTTPS`

2.16.85 SITE_TITLE

Web sitesi ve gönderilecek e-postalar için kullanılacak site başlığı.

2.16.86 SPECIAL_CHARS

Görsel klavyeye eklenecek karakterler. *Sanal klavye*.

Varsayılan değer:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "...")
```

2.16.87 SINGLE_PROJECT

3.8 sürümünde geldi.

Kullanıcıları pano yerine doğrudan bir proje ya da bileşene yönlendirir. `True` olarak ayarlandığında yalnızca Weblate üzerinde tek bir proje olması durumunda çalışır. Alternatif olarak, proje kısaltmasını ayarlarsanız koşulsuz olarak bu projeye yönlendirir.

3.11 sürümünde değişti: Artık tek bir projenin görüntülenmesini dayatmak için bir proje kısaltması da ayarlanabiliyor.

Örnek:

```
SINGLE_PROJECT = "test"
```

2.16.88 SSH_EXTRA_ARGS

4.9 sürümünde geldi.

Weblate SSH çağırırken özel parametreler eklenmesine izin verir. Bu seçenek, eski şifrelemeyi veya diğer standart dışı özellikleri kullanan sunucularla bağlantı kurarken yardımcı olur.

Örneğin, Weblate SSH bağlantısı *Legacyhost ile iletişim kurulamıyor: Uygun bir anahtar değişimi yöntemi bulunamadı*. Önerilen: *diffie-hellman-group1-sha1* hatası ile kurulamadığında şunu kullanarak etkinleştirebilirsiniz:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms =+diffie-hellman-group1-sha1"
```

İpucu: Dizge kabuk tarafından değerlendirilir. Bu nedenle boşluk ve özel karakterleri tırnak arasına aldığınızdan emin olun.

Ayrıca bakınız:

Eski OpenSSH seçenekleri

2.16.89 STATUS_URL

Weblate kopyanızın durumunu bildireceği adres.

2.16.90 SUGGESTION_CLEANUP_DAYS

3.2.1 sürümünde geldi.

Belirtilen sayıda gün sonra önerileri kendiliğinden siler. Varsayılan değer: Yok herhangi bir silme işlemi yapmaz.

2.16.91 UPDATE_LANGUAGES

4.3.2 sürümünde geldi.

Veri tabanı aktarımı çalıştırılırken dil veri tabanının güncellenmesinin gerekip gerekmediğini denetler ve varsayılan olarak etkindir. Bu ayarın :d admin:setuplang ögesinin çağırılması üzerinde herhangi bir etkisi yoktur.

Uyarı: Bu seçenek diller görünümünü tutarsız kılabilir. Weblate dil tanımları zaman içinde zenginleşir ve tanımlanan dillerin dil kodu görüntülenmez.

Ayrıca bakınız:

Hazır dil tanımları

2.16.92 URL_PREFIX

Bu ayar, Weblate platformunun belirli bir yol altında çalışmasını sağlar (yoksa web sunucusunun kök klasöründen çalıştırılır).

Not: Bu ayarı kullanmak için, sunucunuzu bu ön eki kaldıracak şekilde yapılandırmanız gerekir. BU işlem örneğin, WSGI ile WSGIScriptAlias seçeneği ayarlanarak elde edilebilir.

İpucu: Ön ek / karakteri ile başlamalıdır.

Örnek:

```
URL_PREFIX = "/translations"
```

Not: Bu ayar iç Django sunucusuyla çalışmaz. urls.py dosyasını bu ön eki içerecek şekilde ayarlamanız gerekir.

2.16.93 VCS_API_DELAY

4.15.1 sürümünde geldi.

GitHub çekme istekleri, *GitLab birleştirme istekleri*, *Gitea çekme isteği* ve *Pagure birleştirme istekleri* içindeki üçüncü taraf API çağrılarında saniye olarak en az gecikmeyi yapılandırır.

Bu seçenek, aşırı yüklenmelerini önlemek için Weblate üzerinden bu hizmetlere yapılan API çağrılarının hızını sınırlar.

GitHub üzerinde ikincil hız sınırlayıcısına takılıyorsanız, bu değeri artırmak yardımcı olabilir.

Varsayılan değer: 10.

2.16.94 VCS_BACKENDS

Kullanılabilecek sürüm denetimi sistemi arka uçlarının yapılandırması.

Not: Weblate, araçlarının bulunduğu desteklenen tüm arka uçları kullanmaya çalışır.

İpucu: Bu ayar ile seçenekleri sınırlayabilir ya da özel sürüm denetimi sistemi arka uçları ekleyebilirsiniz.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Ayrıca bakınız:

Sürüm denetimi bütünleştirilmesi

2.16.95 VCS_CLONE_DEPTH

3.10.2 sürümünde geldi.

Weblate depolarının hangi alt düzeye kadar kopyalanacağını yapılandırır.

Not: Bu seçenek şu anda yalnızca *Git* için destekleniyor. Varsayılan olarak Weblate, kopyalamayı hızlandırmak ve disk alanını korumak için sığ depo kopyaları oluşturur. Kullanımınıza bağlı olarak (örneğin custom *Eklentiler* kullanırken), kopya derinliği artırmak ya da 0 olarak ayarlayarak tümüyle kapatmak isteyebilirsiniz.

İpucu: Weblate üzerinden iterken ``fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>`` hatası alırsanız, sığ kopyaları tümüyle kapatarak şunları ayarlayın:

```
VCS_CLONE_DEPTH = 0
```

2.16.96 WEBLATE_ADDONS

Kullanılabilecek eklentilerin listesi. Bunların kullanılabilmesi için, belirli bir çeviri bileşeninde etkinleştirilmeleri gerekir. Varsayılan olarak bu, tüm iç eklentileri içerir. Listeyi genişletirken büyük olasılıkla var olanları etkin tutmak isteyeceksiniz. Örneğin:

```
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.LangaugeConsistencyAddon",
    "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",
)
```

Not: Eklentiği listeden kaldırmak, bileşenlerden kaldırmaz. Bu durumda Weblate çöker. Lütfen eklentiği bu listeden kaldırmadan önce tüm bileşenlerden kaldırın.

Ayrıca bakınız:

Eklentiler, *DEFAULT_ADDONS*

2.16.97 WEBLATE_EXPORTERS

4.2 sürümünde geldi.

Çevirilerin veya sözlüklerin çeşitli dosya biçimlerinde indirilmesini sağlayan dışa aktarıcıların listesi.

Ayrıca bakınız:

Desteklenen dosya biçimleri

2.16.98 WEBLATE_FORMATS

3.0 sürümünde geldi.

Kullanılabilecek dosya biçimlerinin listesi.

Not: Varsayılan listede yaygın kullanılan biçimler bulunur.

Ayrıca bakınız:

Desteklenen dosya biçimleri

2.16.99 WEBLATE_MACHINERY

4.13 sürümünde geldi.

Kullanılabilecek makine hizmetlerinin listesi.

Ayrıca bakınız:

Kendiliğinden önerileri yapılandırmak

2.16.100 WEBLATE_GPG_IDENTITY

3.1 sürümünde geldi.

Weblate tarafından Git işlemlerini imzalamak için kullanılacak kimlik. Örneğin:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

Weblate GPG anahtarlığında, uygun bir anahtar aranır (home/.gnupg altında *DATA_DIR*). Bulunamazsa, bir anahtar oluşturulur. Bilgi almak için: *Git işlemlerini GnuPG ile imzalamak*.

Ayrıca bakınız:

Git işlemlerini GnuPG ile imzalamak

2.16.101 WEBSITE_REQUIRED

Bir proje oluşturulurken *Proje web sitesi* ögesinin belirtilmesinin gerekip gerekmediğini belirler. Varyılan olarak, herkese açık sunucu kurulumlarına uygun olarak açıktır.

2.17 Örnek yapılandırma

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"
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# 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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# 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
```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# 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",
            ],
        },
    },
]

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    },
    "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 = (

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "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},
    },
]

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

{ "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",

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

    "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

```

(sonraki sayfaya devam)

```
# 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,
        },
    },
}
```

(önceki sayfadan devam)

```

    # 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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# 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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# 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",

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# "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.LangaugeConsistencyAddon",
#     "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"

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

# 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

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```

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"},
}

```

(sonraki sayfaya devam)

(önceki sayfadan devam)

```
# 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 Yönetim komutları

Not: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check *Dosya sistemi izinleri* 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.

Weblate için virtualenv kullanıyorsanız, **weblate** komutunun tam yolunu belirtebilir ya da çağırmadan önce virtualenv etkinleştirebilirsiniz:

```
# 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
```

Weblate kurulumu için pip kurucusunu ya da `./setup.py` betiğini kullandıysanız, Weblate denetimi için kullanılabacak **weblate** yolunuza (ya da kullandığınız virtualenv yoluna) kurulur:

```
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
```

Ayrıca bakınız:

Docker ile kurmak, Debian ve Ubuntu üzerine kurmak, SUSE ve openSUSE üzerine kurmak, RedHat, Fedora ve CentOS üzerine kurmak, Kaynaklardan kurulum

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

2.5 sürümünde geldi.

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).

Örnek:

```
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>

2.5 sürümünde geldi.

4.6 sürümünde değişti: Çeviri kipi parametresi eklendi.

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 *Tutarsız*).

--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

Specify translation mode, default is `translate` but `fuzzy` or `suggest` can be used.

Örnek:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application_
↪weblate website cs
```

Ayrıca bakınız:

Kendiliğinden çeviri

2.18.4 celery_queues

weblate celery_queues

3.7 sürümünde geldi.

Celery görev kuyruklarının uzunluğunu görüntüler.

2.18.5 checkgit

weblate checkgit <project|project/component>

Arka uç Git deposunun geçerli durumunu yazdırır.

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>

Belirli bir süreden daha önce bekleyen değişiklikleri işler.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

--age HOURS

İşleme için saat olarak yaş. Belirtilmezse, *Bileşen yapılandırması* içinde yapılandırılmış değer kullanılır.

Not: 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 *Bileşen yapılandırması*.

Ayrıca bakınız:

Bakım görevlerini yürütmek, `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.

Ayrıca bakınız:

Bakım görevlerini yürütmek

2.18.9 cleanup_ssh_keys

weblate cleanup_ssh_keys

4.9.1 sürümünde geldi.

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.

Ayrıca bakınız:

SSH depoları

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

Yönetici e-posta adresini belirtin.

--name

Yönetici adını belirtin (görüntülenecek).

--update

Var olan kullanıcıyı güncelleyin (parolaları değiştirmek için bunu kullanabilirsiniz).

2.9 sürümünde değişti: Eklenen parametreler `--username`, `--email`, `--name` ve `--update`.

2.18.11 dump_memory

weblate dump_memory

2.20 sürümünde geldi.

Export a JSON file containing Weblate Translation Memory content.

Ayrıca bakınız:

Çeviri belleği, Weblate çeviri belleği şeması

2.18.12 dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by *importuserdata*.

İpucu: This comes in handy when migrating or merging Weblate instances.

2.18.13 import_demo

weblate import_demo

4.1 sürümünde geldi.

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>

2.7 sürümünde geldi.

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.

2.9 sürümünde değişti: The parameters *--ignore* and *--update* are there to deal with already imported components.

JSON dosyası örneği:


```
[
{
  "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"
}
]
```

Ayrıca bakınız:

import_memory

2.18.15 import_memory

weblate import_memory <file>

2.20 sürümünde geldi.

Imports a TMX or JSON file into the Weblate translation memory.

--language-map LANGMAP

Allows mapping languages in the TMX to the Weblate translation memory. The language codes are mapped after normalization usually done by Weblate.

--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.

Ayrıca bakınız:

Çeviri belleği, Weblate çeviri belleği şeması

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

3.0 sürümünde değişti: The import_project command is now based on the *Bileşen keşfi* 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>[^\]*) / (?P<component>[^-/]*) \.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 *Desteklenen dosya biçimleri*), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see *Bileşen yapılandırması*) 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 \
  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'
```

Ayrıca bakınız:

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

3.2 sürümünde geldi.

```
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

Bir eklentinin JSON olarak kodlanmış yapılandırması.

--update

Var olan eklenti yapılandırmasını güncelle.

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 *Gettext çıktısını özelleştirme* for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --  
↪update --all
```

Ayrıca bakınız:

Eklentiler

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.110n.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

Tüm Weblate bağımlılıklarını ve sürümlerini listeler.

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.

Not: 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.

İpucu: 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.

Ayrıca bakınız:

`unlock_translation`

2.18.25 move_language

weblate move_language source target

3.0 sürümünde geldi.

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.

Örnek:

```
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>

İşlenen değişiklikleri yukarı akış sürüm denetimi deposuna iter.

--force-commit

Bekleyen değişikliklerin itilmeden önce işlenmesini dayatır.

You can either define which project or component to update (for example weblate/application), or use **--all** to update all existing components.

Not: Weblate pushes changes automatically if *İşleme ile itme* in *Bileşen yapılandırması* 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.

İpucu: 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.

Ayrıca bakınız:

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 *Proje erişim denetimi*.

Ayrıca bakınız:

Yetkiler ve hazır rollerin listesi

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.

İpucu: 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.

Not: Usually it is better to configure hooks in the repository to trigger *Bildirim kancaları*, instead of regular polling by *updategit*.

2.19 Duyurular

4.0 sürümünde değişti: Önceki sürümlerde bu özelliğe beyaz tahta iletileri adı veriliyordu.

Site genelinde, projeye, bileşene ya da dile özel duyurular yayınlayarak çevirmenlerinize bilgi sağlayın.

Amacı, bitiş tarihlerini ve durumu duyurun ya da çeviri hedeflerini belirtin.

Kullanıcılar, izledikleri projelerin duyuruları ile ilgili bir bildirim alır (devre dışı bırakmadıkları sürece).

Bu özellik, web sitesinin amacını duyurmaktan çeviriler için hedefler belirlemeye kadar çeşitli konularda yararlı olabilir.

Duyurular, *Yönetim* menüsündeki *Duyuru yayınla* seçeneği ile istenilen düzeyde yayınlanabilir:

Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

🔑

⚠

+

🌐

...

WeblateOrg

translated 90%

Translations will be used only if they reach 60%.

Components

Languages

Info

Search

Insights ▾

Files ▾

Tools ▾

Manage ▾

Share ▾

👁 Not watching ▾

Post announcement

?

Message

You can use Markdown and mention users by @username.

Category

Info (light blue) ▾

Category defines color used for the message.

Expiry date

mm/dd/yyyy

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ Notify users

The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.

Add

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

Yönetim arayüzü kullanılarak da eklenebilir:

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

Duyurular daha sonra belirtilen bağlamlarına göre görüntülenir:

Herhangi bir bağlam belirtilmediğinde

Panoda görüntülenir (giriş sayfası).

Proje belirtildiğinde

Projenin tüm bileşenleri ve çevirilerinde görüntülenir.

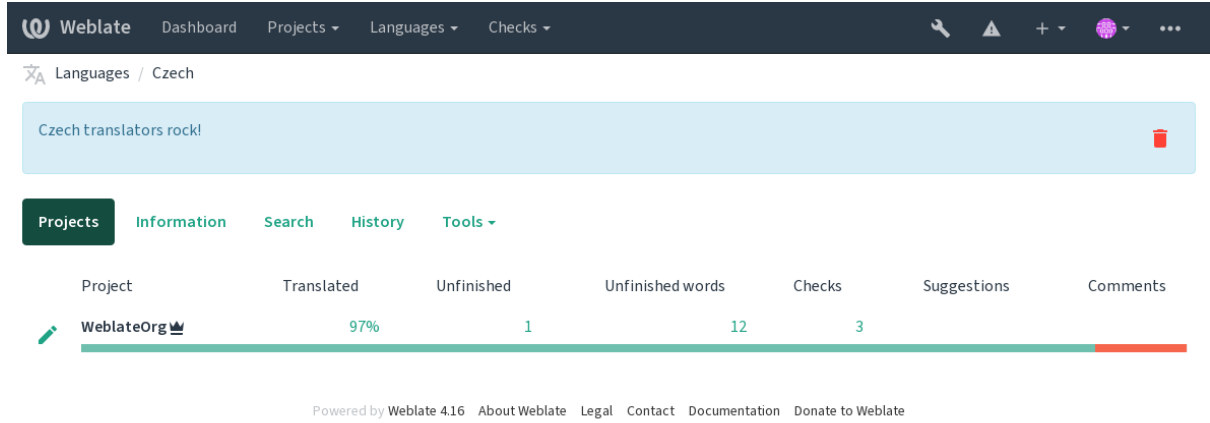
Bileşen belirtildiğinde

Belirtilen bileşen ve tüm çevirilerinde görüntülenir.

Dil belirtildiğinde

Dil özeti ve bu dildeki tüm çevirilerde görüntülenir.

Dil özet sayfası şöyle görüntülenir:



2.20 Bileşen listeleri

Kullanıcı panosunda seçenek olarak görüntülenecek ve kullanıcıların varsayılan görünümüleri olarak birini seçebileceği birkaç bileşen listesi belirtin. Bilgi almak için [Pano](#).

2.20 sürümünde değişti: Panoda sunulan her bileşen listesi için bir durum görüntülenir.

Bileşen listelerinin adları ve içeriği, yönetim arayüzünde, *Bileşen listeleri* bölümünde belirtilebilir. Her bileşen listesinin kullanıcıya görüntülenen bir adı ve adresi ile onu temsil eden bir kısaltması olmalıdır.

2.13 sürümünde değişti: Yönetim bölümünden anonim kullanıcılar için pano ayarlarını değiştirerek kimliği doğrulanmamış kullanıcılara hangi panonun sunulacağını belirtin.

2.20.1 Kendiliğinden bileşen listeleri

2.13 sürümünde geldi.

Kendiliğinden bileşen listesi ataması kuralları oluşturarak bileşenleri kısaltmalarına göre kendiliğinden listeye ekleyebilirsiniz.

- Büyük kurulumlarda bileşen listelerini korumak için ya da Weblate kurulumunuzdaki tüm bileşenleri içeren tek bir bileşen listesi elde etmek istediğinizde kullanışlıdır.

İpucu: Weblate kurulumunuzdaki tüm bileşenlerin bulunduğu bir bileşen listesi oluşturun.

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/WebblateOrg

Chosen components ⓘ

Choose all ⓘ 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="x"/>

[+ Add another Automatic component list assignment](#)

2.21 İsteğe bağlı Weblate modülleri

Several optional modules are available for your setup.

2.21.1 Git exporter

2.10 sürümünde geldi.

Provides you read-only access to the underlying Git repository using HTTP(S).

Kurulum

1. Add `weblate.gitexport` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Export existing repositories by migrating your database after installation:

```
weblate migrate
```

Kullanım

The module automatically hooks into Weblate and sets the exported repository URL in the *Bileşen yapılandırması*. The repositories are accessible under the `/git/` part of the Weblate URL, for example `https://example.org/git/weblate/main/`.

Repositories for publicly available projects can be cloned without authentication:

```
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/'
```

İpucu: By default members or *Users* group and anonymous user have access to the repositories for public projects via *Access repository* and *Power user* roles.

2.21.2 Faturalama

2.4 sürümünde geldi.

This is used on *Hosted Weblate* to define billing plans, track invoices and usage limits.

Kurulum

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
```

Kullanım

After installation you can control billing in the admin interface. Users with billing enabled will get new *Billing* tab in their *Kullanıcı profili*.

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 Yasal

2.15 sürümünde geldi.

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`

Hizmet koşulları belgesi

`legal/documents/privacy.html`

Gizlilik ilkesi belgesi

`legal/documents/summary.html`

Hizmet koşulları ve gizlilik ilkeleri metinlerinin özeti

On changing the terms of service documents, please adjust `LEGAL_TOS_DATE` so that users are forced to agree with the updated documents.

Not: 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.

Kurulum

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.

Kullanım

After installation and editing, the legal documents are shown in the Weblate UI.

2.21.4 Avatarlar

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

Ayrıca bakınız:

Avatar ön belleği, `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:

- Kimliği doğrulanmamış kullanıcılardan öneriler
- Proje ve bileşen açıklamaları ve bağlantıları

Not: This (among other things) relies on IP address of the client, please see *Ters vekil sunucu arkasında çalıştırmak* for properly configuring that.

Ayrıca bakınız:

Ters vekil sunucu arkasında çalıştırmak, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Git işlemlerini GnuPG ile imzalamak


3.1 sürümünde geldi.

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:


[Weblate](#)
[Dashboard](#)
[Projects](#)
[Languages](#)
[Checks](#)
[Register](#)
[Sign in](#)

[About Weblate](#) / [Weblate keys](#)

[About Weblate](#)
[Statistics](#)
[Keys](#)

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
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWdxygRSjmbGwJV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhLwp1uYlW1v
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/WzoBDADL1oNK7oXL4YPx0wZIC99aTB0D5z8gA8Pdlcpk4mkyBveDrLA
D/IW7z/mfgSj0Yx3uZ8gkNMLM9mvAA0zURvQCpvZZN/Cudeqpa7ADtVYeVXaZ
PvCG9M8256PF2ZyO+PUI67HpLEJzHrhFjtjQcRoImKWMWfVfqHsL99lyq2lLeI7q
BaP/RKsbm09ziix/TGNMGPOq4l3JWTFbuApD5yp18235BjuabR++esiy4CtjRe0J
ncpNXAXfyt09IU3gUgmBRE3x8oH3B4fkjNAqK/rRobclWruW/3lsmMr9kt9wjh
lMLevg8aMnd0i+wq48sUpVXoo6nZ5gPoOzIR9EtvEuNJBLoMgQzdg3pgsPhhhH9
SaNbr2+V7BD10nyfqYuZRUec6GcbVXzdXfu8IzSjGktZzfjx14DvprEIL+HsAu2c
9kR08KS12ktUQHQdyJnEjIN5Q2tdabkiNgZwSOGHgjaN9kucnWOYeQDqdyIjklR
fM9xhcf08w6uRZ0AEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWlQRW48B4U8v+rAVEG5xBaEU1QYdagUCY/9bOglbAwULCQgHAgYV
CgklCwIEFgIDAQIeAQIXgAAKCRBxBaEU1QYdapDkDACvKLp0vyAXKzfMNq0hoiyW
IHMk/zGH55mM04V1DyCvMnbuHkMfGu2y8mfq2QsrgeITUBikf0lyLu037+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`.

Ayrıca bakınız:

`WEBLATE_GPG_IDENTITY`

2.21.7 Hız sınırlama

3.2 sürümünde değişti: The rate limiting now accepts more fine-grained configuration.

4.6 sürümünde değişti: 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:

Ad	Kapsam	İzin verilen girişimler	Hız sınırı aralığı	Kilitleme süresi
Hesap açılışı	REGISTRATION	5	300	600
Yöneticilere ileti göndermek	MESSAGE	2	300	600
Oturum açarken parola kimlik doğrulaması	LOGIN	5	300	600
Site genelinde arama	SEARCH	6	60	60
Çeviri	TRANSLATE	30	60	600
Sözlüğe ekleme	GLOSSARY	30	60	600
Yeni bir dil çevirisi başlatma	LANGUAGE	2	300	600
Yeni proje oluşturma	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 hız sınırlaması](#).

Ayrıca bakınız:

[Hız sınırlama](#), [Ters vekil sunucu arkasında çalıştırmak](#), [API hız sınırlaması](#)

2.21.8 Fedora iletileri bütünleştirmesi

Fedora Messaging, Weblate üzerinde yapılan tüm değişiklikleri ileten AMQP temelli bir yayınlıyıcıdır. Bunu kullanarak Weblate üzerindeki değişikliklere ek hizmetler bağlayabilirsiniz.

Fedora Messagin bütünleştirmesi, `weblate-fedora-messaging` ile ayrı bir Python modülü olarak kullanılabilir. Kurulum yönergeleri için https://github.com/WeblateOrg/fedora_messaging/ adresine bakabilirsiniz.

2.22 Weblate platformunu özelleştirmek

Django ve Python kullanarak zenginleştirebilir ve özelleştirebilirsiniz. Herkesin yararlanabilmesi için değişikliklerinizi yukarı akışa katın. Bu yöntem, bakım maliyetlerinizi azaltır. İç arayüzler değiştirilirken veya kod yeniden düzenlenirken Weblate üzerindeki kod ele alınır.

Uyarı: Ne iç arayüzler ne de kalıplar kararlı bir API olarak kabul edilmez. Lütfen her yükseltme için kendi özelleştirmelerinizi gözden geçirin. Arayüzler veya bağlamları önceden bildirilmeksizin değişebilir.

Ayrıca bakınız:

[Weblate için katkıda bulunmak](#)

2.22.1 Bir Python modülü oluşturmak

Python ile ilgili değilseniz, temel bilgileri açıklayan ve eğitimlerin bulunduğu ‘Yeni başlayanlar için Python <<https://www.python.org/about/gettingstarted/>>’ bölümüne bakmak isteyebilirsiniz.

Özel Python kodu (modül olarak adlandırılır) içeren bir dosya yazmak için, sistem yolunda (genellikle `/usr/lib/python3.9/site-packages/` gibi bir şey) veya yorumlayıcı arama yoluna da eklenen Weblate klasöründe bir depolama alanına gerek duyulur.

3.8-5 sürümünde geldi: *Docker kullanıldığında*, Python modüllerini `/app/data/python/` (see *Docker kapsayıcısı birimleri*) içine yerleştirebilirsiniz. Böylece Weblate bunları yükleyebilir. Örneğin bir *ayar değiştirme dosyası* kullanabilirsiniz.

Daha da iyisi, özelleştirmenizi uygun bir Python paketine dönüştürebilirsiniz:

1. Paketiniz için bir klasör oluşturun (`weblate_customization` kullanacağız).
2. İçinde paketi açıklayan bir `setup.py` dosyası oluşturun:

```
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. Özelleştirme kodunun bulunduğu Python modülü için bir klasör oluşturun (`weblate_customization` olarak da adlandırılır).
4. İçinde bir `__init__.py` dosyası oluşturarak modülün Python tarafından içe aktarabildiğinden emin olun.
5. Bu paket artık `pip install -e` kullanılarak kurulabilir. Bilgi almak için: [Editable installs](#).
6. Kurulduktan sonra, modül Weblate yapılandırmasında kullanılabilir (`weblate_customization.checks.FooCheck` gibi).

Paket yapınız şöyle görünmelidir:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

<<https://github.com/WeblateOrg/customize-example>> adresinde bir Weblate özelleştirme örneği bulabilirsiniz. Bu örnek aşağıda açıklanan tüm konuları kapsar.

2.22.2 Logoyu değiştirmek

1. Üzerine yazmak istediğiniz durağan dosyaların bulunduğu basit bir Django uygulaması oluşturun (bilgi almak için: *Bir Python modülü oluşturmak*).

Marka şu dosyalarda bulunur:

icons/weblate.svg

Gezinme çubuğunda görüntülenecek logo.

logo-*.png

Ekran çözünürlüğüne ve web tarayıcısına bağlı olarak web simgeleri.

favicon.ico

Eski tarayıcılar tarafından kullanılan web simgesi.

weblate-*.png

Botlar ve anonim kullanıcılar için avatarlar. Bazı web tarayıcıları bunları kısayol simgeleri olarak kullanır.

email-logo.png

Bildirim e-postalarında kullanılır.

2. `INSTALLED_APPS` içine ekleyin:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here...
)
```

3. İstemcilere sunulacak durağan dosyaları derlemek için `weblate collectstatic --noinput` komutunu yürütün.

Ayrıca bakınız:

How to manage static files (e.g. images, JavaScript, CSS), *Durağan dosyalar sunmak*

2.22.3 Özel kalite denetimleri, eklentiler ve kendiliğinden düzeltmeler

Weblate üzerine *Özel kendiliğinden düzeltmeler*, *Kendi denetimlerinizi oluşturmak* ya da *Eklenti yazma* kodunuzu kurmak için:

1. Dosyaları Weblate özelleştirmesinin bulunduğu Python modülünüze yerleştirin (bilgi almak için: *Bir Python modülü oluşturmak*).
2. Python sınıfının tam yolunu özel ayarlara ekleyin (`WEBLATE_ADDONS`, `CHECK_LIST` ya da `AUTO-FIX_LIST`):

```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Ayrıca bakınız:

Özel kendiliğinden düzeltmeler, *Kendi denetimlerinizi oluşturmak*, *Eklenti yazma*, *Eklentiden betikleri çalıştırma*

2.23 Yönetim arayüzü

Yönetim bölümü, `/manage/` adresinden yönetim ayarlarının yapılmasını sağlar. Yönetici yetkileri ile oturum açmış kullanıcılar tarafından kullanılabilir ve sağ üstteki İngiliz anahtarı simgesi kullanılarak erişilebilir:

The screenshot shows the Weblate management interface. At the top, there's a navigation bar with 'Weblate' logo, 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a 'Manage' section with a sidebar containing 'Weblate status', 'Backups', 'Translation memory', 'Performance report', 'SSH keys', 'Alerts', 'Repositories', 'Users', and 'Teams'. The 'Weblate status' section is active, showing the 'Weblate version' as 4.16 with a unique hash, 'Support status' as 'Community support', and buttons for 'Purchase support package' and 'Donate to Weblate'. Below this is the 'Activate support package' section, which includes a text area for an 'Activation token' and buttons for 'Activate' and 'Purchase support package'.

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

Weblate hakkında temel bilgileri içerir:

- Destek durumu. Bilgi almak için [Weblate için destek almak](#)
- Yedekler. Bilgi almak için [Weblate yedeğini alma ve taşıma](#)
- Paylaşılan çeviri belleği. Bilgi almak için [Çeviri belleği](#)
- Weblate sağlığı ve Celery kuyruklarının uzunluğunu gösteren başarımlar raporu
- SSH anahtar yönetimi. Bilgi almak için [SSH depoları](#)
- Tüm bileşenlerin uyarı özeti. Bilgi almak için alerts

2.23.1 Django yönetim arayüzü

Uyarı: Bu arayüz düşük düzeyde olduğundan kullanırken dikkatli olun. Genellikle bunu kullanmaya gerek duymamalıdır. Çoğu şeye Weblate kullanıcı arayüzü veya API üzerinden rahatça ulaşılabilir.

Bu bölümden kullanıcılar, çeviriler gibi veri tabanında depolanan nesneleri ve diğer ayarları yönetebilirsiniz:

Webplate administration

WELCOME WEBLATE TEST RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT

Site administration

REPORTS

Webplate support status

Status of repositories

SSH keys

Performance report

Translation memory

ACCOUNTS

Audit log entries

+ Add

Change

User profiles

+ Add

Change

Verified e-mails

+ Add

Change

AUTH TOKEN

Tokens

+ Add

Change

AUTHENTICATION

Groups

+ Add

Change

Roles

+ Add

Change

Users

+ Add

Change

BILLING

Billing plans

+ Add

Change

Customer billings

+ Add

Change

Invoices

+ Add

Change

FONTS

Font groups

+ Add

Change

Fonts

+ Add

Change

LEGAL

TOS agreements

+ Add

Change

PERIODIC TASKS

Clocked

+ Add

Change

Crontabs

+ Add

Change

Intervals

+ Add

Change

Periodic tasks

+ Add

Change

Solar events

+ Add

Change

PYTHON SOCIAL AUTH

Associations

+ Add

Change

Nonces

+ Add

Change

User social auths

+ Add

Change

SCREENSHOTS

Screenshots

+ Add

Change

TRANSLATION MEMORY

Translation memory entries

+ Add

Change

WEBLATE CONFIGURATION

Settings

+ Add

Change

WEBLATE LANGUAGES

Languages

+ Add

Change

WEBLATE TRANSLATIONS

Announcements

+ Add

Change

Component lists

+ Add

Change

Components

+ Add

Change

Contributor agreements

+ Add

Change

Projects

+ Add

Change

Recent actions

My actions

None available

In the *Raporlar* bölümünden, sitenizin durumuna bakabilirsiniz. *Üretim kurulumu* kipine geçebilir ya da *Depolara erişmek* erişimi için SSH anahtarlarını yönetebilirsiniz.

Bölümlerden herhangi birinin altındaki veri tabanı nesnelerini yönetin. Büyük olasılıkla en ilginç olanı, çevrilebilir projeleri yönetebileceğiniz *Weblate çevirileri*. Bilgi almak için: *Proje yapılandırması* ve *Bileşen yapılandırması*.

Weblate dilleri bölümünde dil tanımları bulunur. Bilgi almak için: *Dil tanımları*.

Proje ekleme

Eklenen bir proje, tüm bileşenler için bir kapsayıcı görevi görür. Genellikle bir yazılım ya da kitap parçası için bir proje oluşturursunuz (kullanılacak parametreler hakkında bilgi almak için: *Proje yapılandırması* bölümüne bakabilirsiniz):

Webate administration

WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

[Home](#) · [Webate translations](#) · [Projects](#) · [Add Project](#)

Add Project

Required fields are marked in bold.

Project name:

WebateOrg

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.

☒ 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

Ayrıca bakınız:

Proje yapılandırması

Çift dilde bileşenler

Bir proje ekledikten sonra, ona çeviri bileşenleri ekleyebilirsiniz. (kullanılabilecek parametrelerle ilgili bilgi almak için *Bileşen yapılandırması* bölümüne bakabilirsiniz):

412

Bölüm 2. Yönetici belgeleri

Ayrıca bakınız:

Bileşen yapılandırması, İki dilli ve tek dilli biçimler

Tek dilde bileşenler

Bunların daha kolay çevrilmesi için, ileti kimliklerinin ilgili kaynak diliyle (genellikle İngilizce) eşlenebileceği bir kalıp dosyası sağlayın (kullanılabilecek parametrelerle ilgili bilgi almak için *Bileşen yapılandırması* bölümüne bakabilirsiniz):

414

Bölüm 2. Yönetici belgeleri

Ayrıca bakınız:*Bileşen yapılandırması, İki dilli ve tek dilli biçimler*

2.24 Weblate için destek almak

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 Destek bütünleştirmesi

3.8 sürümünde geldi.

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 shows the Weblate web interface. At the top, there's a navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with several tabs: Weblate status (selected), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, Teams, Appearance, Tools, Automatic suggestions, and Billing. The 'Weblate status' section displays the following information:

- Weblate support status** (with an info icon)
- Weblate version**: 4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5
- Support status**: Community support (with a 'Refresh support status' link)
- Buttons: Purchase support package, Donate to Weblate

Below this is the 'Activate support package' section (with an info icon). It contains the text: 'The support packages include priority e-mail support, or cloud backups of your Weblate installation.' Underneath is the 'Activation token' section with an input field and the instruction: 'Please enter the activation token obtained when making the subscription.' At the bottom of this section are buttons for 'Activate' and 'Purchase support package'.

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

2.24.2 Weblate üzerine gönderilen veriler

- Weblate kopyanızın yapılandırıldığı adres
- Sitenizin başlığı
- Kullandığınız Weblate sürümü
- Weblate veri tabanınızdaki bazı nesnelerin günlükleri (projeler, bileşenler, diller, kaynak dizgeler ve kullanıcılar)
- Kopyanızın herkese açık SSH anahtarı

Ayrıca, *Weblate uygulamasını keşfedin* açık olduğunda:

- Herkese açık projelerin listesi (ad, adres ve web sitesi)

Başka herhangi bir veri gönderilmez.

2.24.3 Bütünleştirme hizmetleri

- Destek paketinizin hala geçerli olup olmadığını görmek
- *Weblate tarafından sunulan yedekleme depolama alanı*
- *Weblate uygulamasını keşfedin*

İpucu: Satın alınan destek paketleri o sırada etkinleştirilir ve bütünleştirilmeden kullanılabilir.

2.24.4 Weblate uygulamasını keşfedin

4.5.2 sürümünde geldi.

Not: Bu özellik şu anda erken beta aşamasındadır.

Discover Weblate, kullanıcıların Weblate sunucularını ve topluluklarını bulmasını kolaylaştıran bir abonelik hizmetidir. Kullanıcılar <<https://weblate.org/discover/>> adresinden kaydedilmiş hizmetlere göz atabilir ve katkıda bulunulacak projeleri bulabilir.

Listelenme

İpucu: Discover Weblate hizmetine katılmak, Weblate kopyanızın sunucunuz hakkında bazı bilgiler göndermesini sağlar. Bilgi almak için: *Weblate üzerine gönderilen veriler*.

Sunucunuzun etkin bir destek aboneliği ile Discovery Weblate hizmetinde listelenmesi için (bilgi almak için: *Destek bütünleştirilmesi*) yönetim panosundan şu seçeneği açmanız yeterlidir:

Weblate
 Dashboard Projects 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](#) [Browse discovery](#)

[Enable 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](#)

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

Sunucunuzu destek aboneliği olmadan Discover Weblate hizmetinde listeleme:

1. <https://weblate.org/user/> adresinden kendinizi kaydedin
2. Weblate sunucunuzu <https://weblate.org/subscription/discovery/> adresindeki Discover Weblate veri tabanına kaydedin
3. Weblate kopyanızda hizmetin etkinleştirmesini onaylayın ve *Weblate Discovery kullanılsın* düğmesini kullanarak Weblate yönetimi sayfanızda keşif listesini açın:



 Weblate


Dashboard

Projects ▾

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 version4.16 — [eb73d693310f0d9403c31ded4ff00c5f778150b5](#)

Support statusCommunity support

Refresh support status

Discover WeblateYour 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

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

Listelemeyi özelleştirme

`<https://weblate.org/user/>` adresinden bir metin ve görsel (570 x 260 piksel) belirterek listelemeyi özelleştirebilirsiniz.

2.25 Yasal belgeler

Not: 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.

418

Bölüm 2. Yönetici belgeleri

2.25.1 Lisans uyumluluğu

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 ABD şifreleme denetimleri

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

Bilgi almak için: https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG

Weblate tarafından isteğe bağlı olarak kullanılır

Git

Weblate tarafından isteğe bağlı olarak kullanılır

curl

Git tarafından kullanılır

OpenSSL

Python ve cURL tarafından kullanılır

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)

Ayrıca bakınız:

[Export Controls \(EAR\) on Open Source Software](#)

3.1 Weblate için katkıda bulunmak

Weblate platformunu geliştirmenin pek çok yolu vardır. Kodlama, görsel tasarım, belge hazırlama, destek olma veya fikir verme gibi size uygun olan birini seçebilirsiniz:

- *Weblate sorunlarını bildirmek*
- *Weblate kodlarına katkıda bulunmak*
- *Weblate modüllerine katkıda bulunmak*
- *Weblate arayüzünü çevirme*
- *Weblate belgelerine katkıda bulunun*
- *Weblate tartışmaları*
- *Weblate geliştirilme çalışmalarına maddi destek vermek*

3.1.1 Weblate arayüzünü çevirme

Weblate, gene Weblate kullanılarak sürekli olarak [çevriliyor](#). Weblate kullanımının olabildiğince çok dilde sağlanmasına yardımcı olmaktan çekinmeyin. Böylece Weblate kullanıcıları ile yakınlaşır!

Kaynak dizgede olası bir hata bulursanız, bunu Weblate düzenleyicisinde bir yorumla belirtebilirsiniz. Böylece üzerinde tartışılabilir ve düzeltilebilir. Eminseniz, *Kaynak dizge konumu* bölümündeki bağlantıya tıklayabilir ve düzeltmenizle birlikte bir çekme isteği gönderebilirsiniz.

3.1.2 Weblate belgelerine katkıda bulunun

İstedığınız belgeler sayfasını iyileştirebilirsiniz. Bunun için sayfanın sağ üst köşesindeki *GitHub üzerinde düzenle* düğmesine tıklayarak kolayca yapabilirsiniz.

Lütfen yazarken şu yönergelerle uyun:

1. Geçerliyse, belgenin bir bölümünü kaldırmayın.
2. Açık ve kolay anlaşılır bir dil kullanın. Teknoloji belgeleri yazıyorsunuz, şiir değil. Belgeleri okuyacak herkesin ana dili İngilizce değildir, düşünceli olun.
3. Emin değilseniz sormaktan çekinmeyin. Düzenleme yaparken bazı özellikler hakkında soru sormanız gerekiyorsa, yanıt almadan önce belgeleri değiştirmeyin. Özetle: Değiştirin ya da sorun. İkisini aynı anda yapmayın.
4. Belgeleri izlerken açıklanan işlemleri yaparak değişikliklerinizi doğrulayın.
5. Onaylama ve birleştirmenin daha kolay ve hızlı yapılması için çekme isteğinde küçük değişiklik parçaları gönderin.
6. Büyük bir makaleyi yeniden yazmak ve yapısını değiştirmek istiyorsanız, bunu iki adımda yapın:
 1. Yeniden yazın
 2. Yeniden yazma onaylandıktan, iyileştirildikten ve birleştirildikten sonra, başka bir çekme isteğinde paragrafların yapısını değiştirin.

İpucu: Belgeleri çevirebilirsiniz.

3.1.3 İç dil tanımlarını zenginleştirmek

Dil tanımları `weblate-language-data` deposunda bulunur.

Eksik dil tanımlarını `languages.csv` dosyasına ekleyebilirsiniz. Diğer dosyalar bu dosyadan oluşturulur.

3.1.4 Weblate tartışmaları

Bir fikriniz varsa ve bir sorun olarak bildirmeye uygun olup olmadığından emin değilseniz endişelenmeyin. [GitHub tartışmaları](#) ile topluluğa katılıp tartışabilirsiniz.

3.1.5 Weblate geliştirilme çalışmalarına maddi destek vermek

Weblate geliştirme çalışmalarını [bağış sayfasından](#) destekleyebilirsiniz. Buradan toplanan fonlar, özgür yazılım projeleri için ücretsiz barındırma hizmeti sağlanması ve Weblate özelliklerinin geliştirilmesi için kullanılır. Fonlama hedefleri ile gururlu bir fon sağlayıcı olarak alacağınız ödüller gibi konular için [`bağış sayfasına`](#) bakabilirsiniz.

Weblate için bağış yapan destekçiler

Weblate destekçilerinin listesi:

- 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

Bu listede yer almak ister misiniz? [Weblate için bağış](#) seçeneklerine bakabilirsiniz.

3.2 Weblate kodlarına katkıda bulunmak

Weblate kaynak kodlarını anlamak için lütfen önce *Weblate kaynak kodları*, *Weblate frontend* ve *Weblate iç yapısı* bölümlerine bakın.

3.2.1 Kod tabanı ile başlamak

Weblate kod tabanına alışmak için bazı hatalar arıyorsanız, [good first issue](#) olarak etiketlenmiş olanlara bakın.

Bu konular üzerinde sormadan çalışmaya başlayabilirsiniz. Yalnızca çalışmaya başladığınızı duyurun. Böylece herkes birisinin bu konu üzerinde çalıştığını bilir.

3.2.2 Weblate platformunu yerel olarak çalıştırma

Weblate geliştirmeye başlamanın en kolay yolu *Kaynaklardan kurulum* bölümündeki yönergeleri izlemektir. Böylece, düzenlenebilir Weblate kaynaklarını içeren bir virtualenv oluşturabilirsiniz.

1. Weblate kaynak kodlarını kopyalayın:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Bir virtualenv oluşturun:

```
virtualenv .venv
.venv/bin/activate
```

3. Weblate kurulumunu yapın (bunun için bazı sistem bağımlılıklarına gerek duyulur. Bilgi almak için: *Kaynaklardan kurulum*):

```
pip install -e .
```

3. Geliştirme için yararlı olacak tüm bağımlılıkları kurun:

```
pip install -r requirements-dev.txt
```

4. Bir geliştirme sunucusu başlatın:

```
weblate runserver
```

5. Yapılandırmanıza bağlı olarak, Celery işlemlerini de başlatmak isteyebilirsiniz:

```
./weblate/examples/celery start
```

6. Bir deneme yapın (bilgi almak için: *Local testing*):

```
. scripts/test-database.sh
./manage.py test
```

Ayrıca bakınız:

Kaynaklardan kurulum

3.2.3 Weblate platformunu Docker üzerinde yerel olarak çalıştırma

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.

The script also accepts some parameters, to execute tests, run it with the `test` parameter and then specify any `test` parameters, for example running only tests in the `weblate.machine` module:

```
./rundev.sh test --failfast weblate.machine
```

Not: 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
```

Running the script without arguments will re-create the Docker container and restart it.

Not: This is not a suitable setup for production, as it includes several hacks which are insecure, but they make development easier.

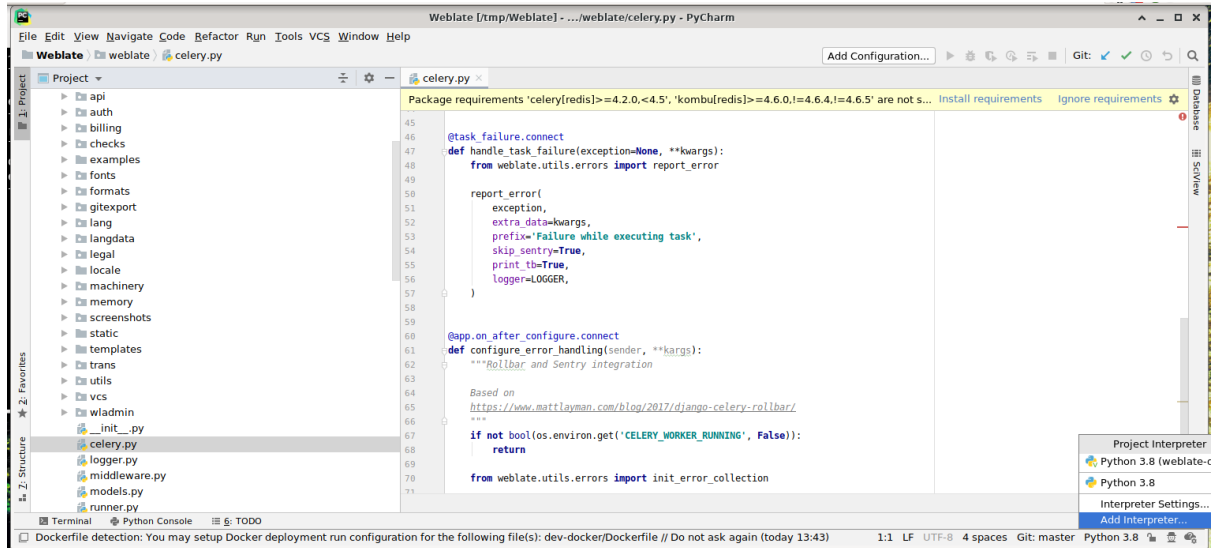
3.2.4 Bootstrapping your devel instance

You might want to use `import_demo` to create demo translations and `createadmin` to make an admin user.

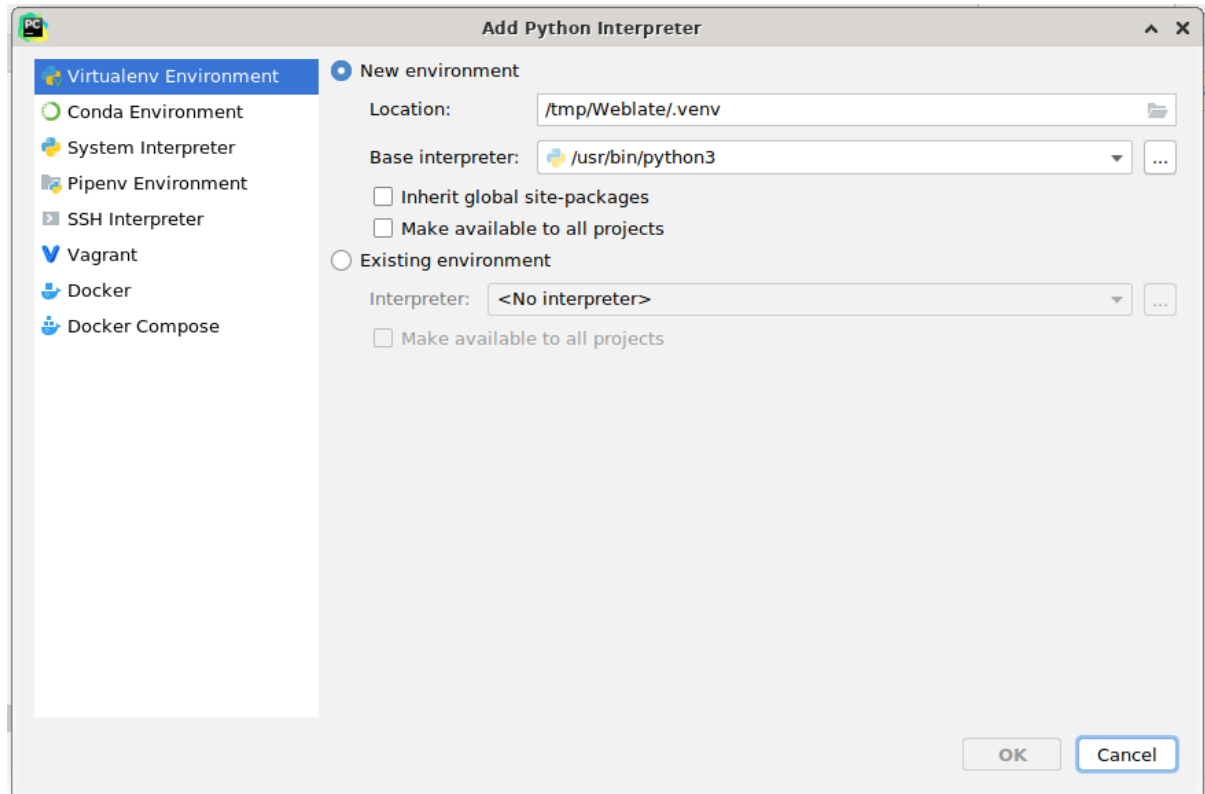
3.2.5 Coding Weblate with PyCharm

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:

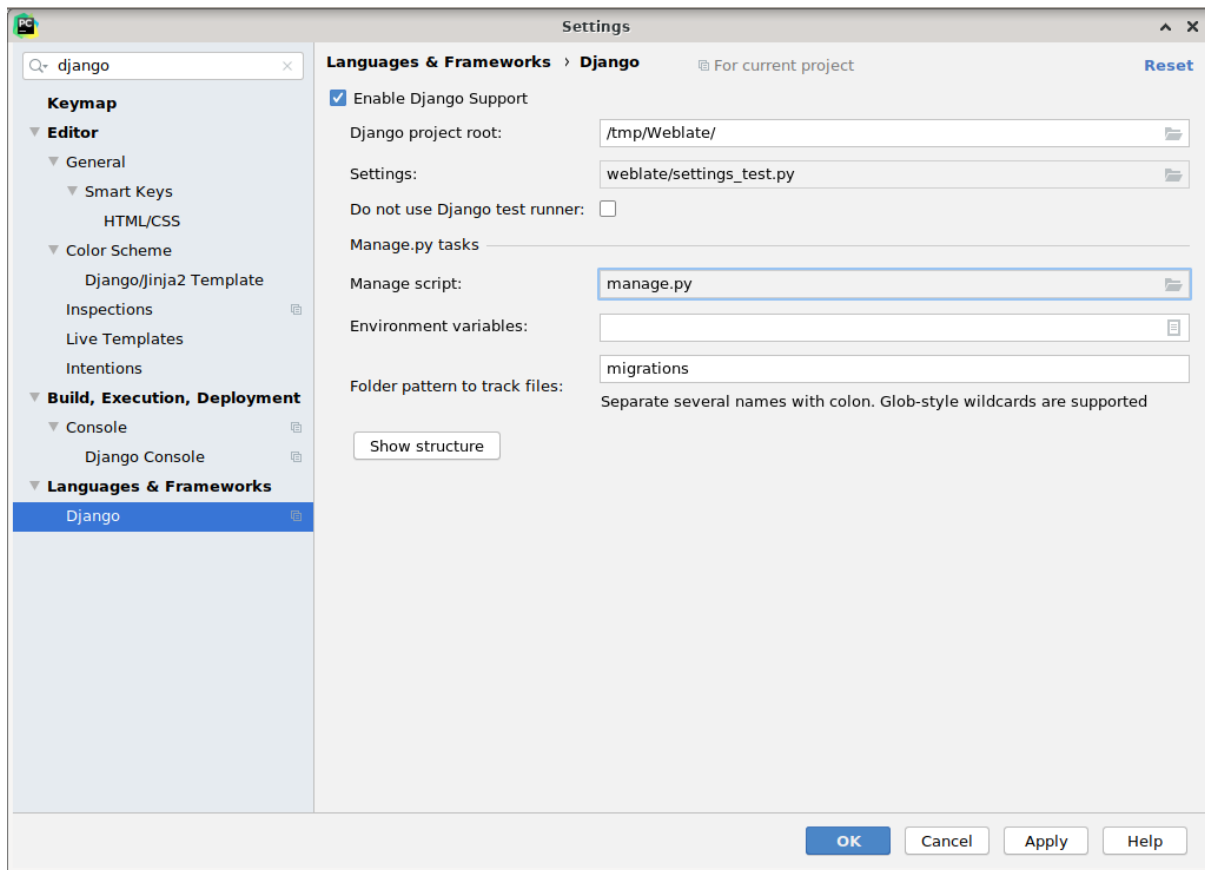


PyCharm sizin için bir virtualenv oluşturabilir ya da zaten var olan bir virtualenv kullanmayı seçebilirsiniz:



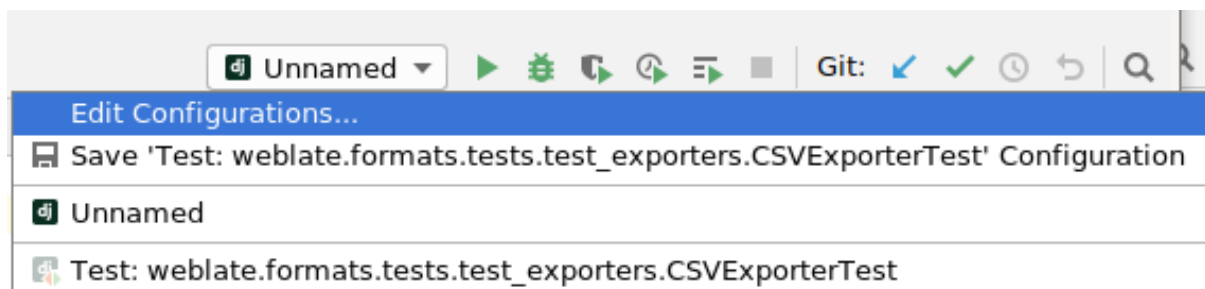
Yorumlayıcı ayarlandıktan sonra bağımlılıkları kurmayı unutmayın: Konsol üzerinden (IDE içindeki konsol varsayılan olarak doğrudan belirttiğiniz virtualenv kullanır) ya da eksik bağımlılıklar hakkında bir uyarı aldığınızda arabirim üzerinden.

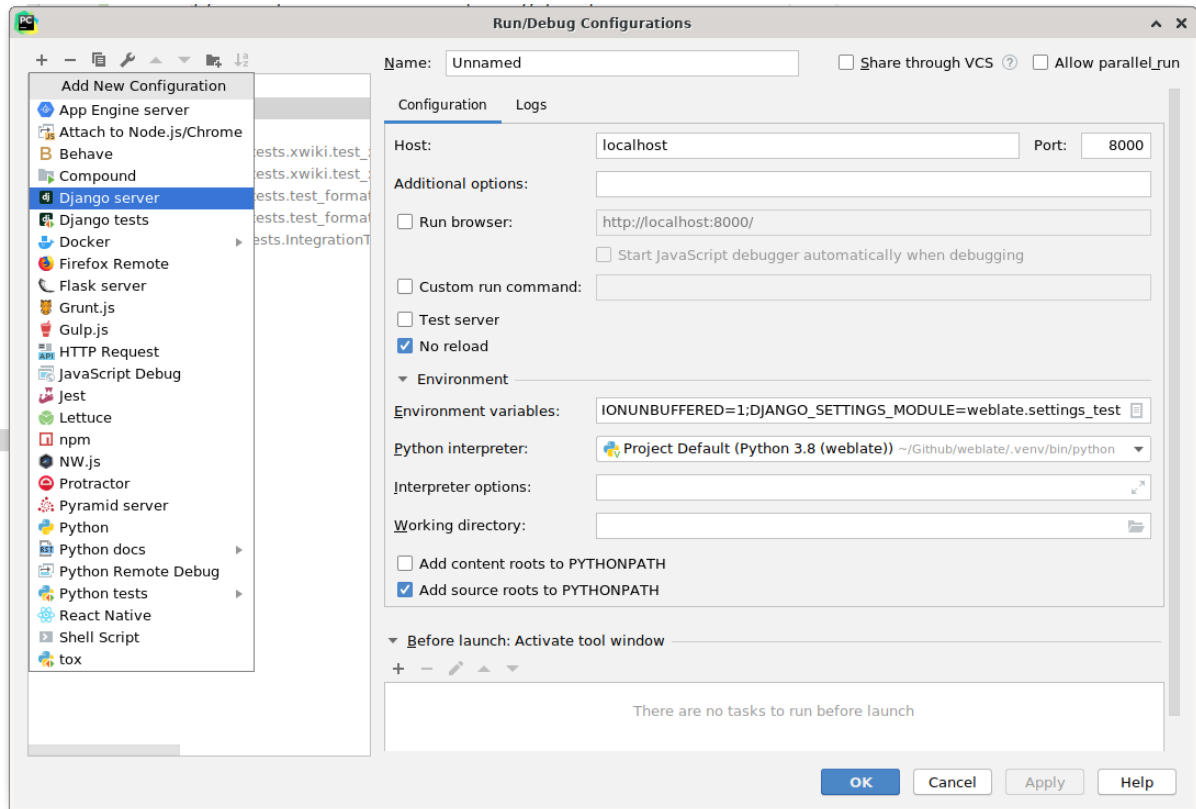
The second step is to set the right info to use Django natively inside PyCharm: The idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of the Django project and the path to its settings:



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:





İpucu: Be careful with the property called *No reload*: It prevents the server from being reloaded live if you modify files. This allows the existing debugger breakpoints to persist, when they normally would be discarded upon reloading the server.

3.3 Weblate kaynak kodları

Weblate, [GitHub](#) üzerinde geliştirilmektedir. Kodu çatallayabilir ve çekme istekleri yapabilirsiniz. Başka herhangi bir biçimdeki yamalar da kabul edilir.

Ayrıca bakınız:

Weblate platformunun içeriden nasıl görüldüğünü anlamak için [Weblate iç yapısı](#) bölümüne bakabilirsiniz.

3.3.1 Kodlama rehberi

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 Debugging Weblate

Bugs can behave as application crashes or as various misbehavior. You are welcome to collect info on any such issue and submit it to the [issue tracker](#).

3.4.1 Hata ayıklama kipi

Turning on debug mode will make the exceptions show in the web browser. This is useful to debug issues in the web interface, but not suitable for a production environment because it has performance consequences and might leak private data.

In a production environment, use [ADMINS](#) to receive e-mails containing error reports, or configure error collection using a third-party service.

Ayrıca bakınız:

Hata ayıklama kipini kapatın, Yöneticileri düzgün şekilde yapılandırın, Hata raporlarını derlemek

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 [Celery ile arka plan görevlerini kullanmak](#)) 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`.

Ayrıca bakınız:

Örnek yapılandırma contains `LOGGING` configuration.

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 a related issue.

Bu durumda denetlenecek olanlar:

- Check that the Celery process is running, see *Celery ile arka plan görevlerini kullanmak*
- Check the Celery queue status, either in *Yönetim arayüzü*, or using `celery_queues`
- Look in the Celery logs for errors (see *Weblate logs*)

3.4.4 Not receiving e-mails from Weblate

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 *Yönetim arayüzü* under the *Tools* tab.

These send e-mails directly, so this verifies that your SMTP configuration is correct (see *Giden e-postayı yapılan-dırmak*). 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 Analyzing application crashes

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 *Hata raporlarını derlemek*.

3.4.6 Silent failures

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 *Hata raporlarını derlemek* helps you to notice such failures easier.

3.4.7 Performance issues

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 *Hata raporlarını derlemek* and get pinpointed and detailed tracebacks in the error collection tool.

In case the slow performance is linked to the database, you can also enable logging of all database queries using following configuration after enabling `DEBUG`:

```
LOGGING["loggers"]["django.db.backends"] = {"handlers": ["console"], "level":  
↪ "DEBUG" }
```


3.5 Weblate iç yapısı

Not: Bu bölüm size Weblate iç yapısına temel bir genel bakış sunar.

Weblate, kod yapısının çoğu [Django](#) temelinden türetilmiştir.

3.5.1 Klasör yapısı

Weblate ana deposunun klasör yapısına hızlı bir bakış:

docs

Bu belgelerin ‘Sphinx’ <<https://www.sphinx-doc.org/>>_ kullanılarak oluşturulabilen kaynak kodu.

dev-docker

Geliştirme sunucusunu çalıştırmak için Docker kodu. Bilgi almak için: *Weblate platformunu Docker üzerinde yerel olarak çalıştırma*.

weblate

Bir [Django](#) uygulaması olarak Weblate kaynak kodu. Bilgi almak için: *Weblate iç yapısı*.

weblate/static

İstemci dosyaları (CSS, JavaScript ve görseller). Bilgi almak için: *Weblate frontend*.

3.5.2 Modüller

Weblate birkaç Django uygulamasından oluşur (bazıları isteğe bağlıdır). Bilgi almak için: *İsteğe bağlı Weblate modülleri*:

accounts

Kullanıcı hesabı, profiller ve bildirimler.

addons

Weblate davranışını değiştiren eklentiler. Bilgi almak için: *Eklentiler*.

api

[Django REST çatısı](#) üzerine geliştirilmiş API.

auth

Kimlik doğrulama ve izinler.

billing

İsteğe bağlı *Faturalama* modülü.

checks

Çeviri dizgesi: ref: *checks* modülü.

fonts

Yazı tipi görüntüleme denetimleri modülü.

formats

Translate-toolkit temelli dosya biçimi soyutlama katmanı.

gitexport

İsteğe bağlı *Git exporter* modülü.

lang

Dil ve çoğul modelleri tanımlayan modül.

legal

İsteğe bağlı *Yasal* modülü.

machinery

Makine çevirisi hizmetleri bütünleştirmesi.

memory

İç çeviri belleği, bkz: *Çeviri belleği*.

screenshots

Ekran görüntüsü yönetimi ve optik karakter tanıma (OCR) modülü.

trans

Çevirileri işleyen ana modül.

utils

Çeşitli yardımcı programlar.

vcs

Sürüm denetimi sistemi soyutlaması.

wladmin

Django yönetim arayüzü özelleştirmesi.

3.6 Eklentileri geliştirmek

Weblate üzerinde yerelleştirme iş akışını özelleştirmek için *Eklentiler* kullanılabilir.

class weblate.addons.base.BaseAddon (*storage = None*)

Weblate eklentileri için temel sınıf.

classmethod can_install (*component, user*)

Eklentinin belirtilen bileşenle uyumlu olup olmadığını denetler.

configure (*settings*)

Yapılandırmayı kaydeder.

daily (*component*)

Kanca günlük olarak tetiklenir.

classmethod get_add_form (*user, component, **kwargs*)

Yeni eklenti eklemek için yapılandırma formunu görüntüler.

get_settings_form (*user, **kwargs*)

Bu eklenti için yapılandırma formunu görüntüler.

post_add (*translation*)

Kanca yeni çeviri eklendikten sonra tetiklenir.

post_commit (*component*)

Kanca değişiklikler depoya işlendikten sonra tetiklenir.

post_push (*component*)

Kanca depo yukarı akışa yollandıktan sonra tetiklenir.

post_update (*component, previous_head: str, skip_push: bool*)

Kanca depo yukarı akıştan güncellendikten sonra tetiklenir.

Parametreler

- **previous_head** (*str*) – Güncellemeden önce deponun HEAD bölümü. İlk kopyalamada boş olabilir.
- **skip_push** (*bool*) – Eklenti işleminin değişiklikleri yukarı akışa itmeyi atlayıp atlamayacağı. Genellikle bunu temeldeki yöntemlere `commit_and_push` ya da `commit_pending` olarak iletebilirsiniz.

pre_commit (*translation, author*)

Kanca değişiklikler depoya işlenmeden önce tetiklenir.

pre_push (*component*)

Kanca depo yukarı akışa yollanmadan önce tetiklenir.

pre_update (*component*)

Kanca depo yukarı akıştan güncellenmeden önce tetiklenir.

save_state ()

Eklenti durum bilgilerini kaydeder.

store_post_load (*translation, store*)

Kanca bir dosya işlendikten sonra tetiklenir.

Parametre olarak bir dosya biçimi sınıfının bir kopyasını alır.

Bu seçenek, dosyanın nasıl kaydedileceğini ayarlamak gibi dosya biçimi sınıfı parametrelerini değiştirmek için faydalıdır.

unit_pre_create (*unit*)

Kanca yeni birim oluşturulmadan önce tetiklenir.

İşte örnek bir eklenti:

```
# 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

The frontend is currently built using Bootstrap, jQuery and few third party libraries.

3.7.1 Desteklenen tarayıcılar

Weblate supports the latest, stable releases of all major browsers and platforms.

Alternative browsers which use the latest version of WebKit, Blink, or Gecko, whether directly or via the platform's web view API, are not explicitly supported. However, Weblate should (in most cases) display and function correctly in these browsers as well.

Older browsers might work, but some features might be limited.

3.7.2 Dependency management

The yarn package manager is used to update third party libraries. The configuration lives in `scripts/yarn` and there is a wrapper script `scripts/yarn-update` to upgrade the libraries, build them and copy to correct locations in `weblate/static/vendor`, where all third partly frontend code is located. The Weblate specific code should be placed directly in `weblate/static` or feature specific subdirectories (for example `weblate/static/editor`).

Adding new third-party library typically consists of:

```
# 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 Kodlama biçemi

Weblate relies on [Prettier](#) for the code formatting for both JavaScript and CSS files.

We also use [ESLint](#) to check the JavaScript code.

3.7.4 Yerelleştirme

Should you need any user visible text in the frontend code, it should be localizable. In most cases all you need is to wrap your text inside `gettext` function, but there are more complex features available:

```
document.write(gettext('this is to be translated'));

var object_count = 1 // or 0, or 2, or 3, ...
s = gettext('literal for the singular case',
            'literal for the plural case', object_count);

fmts = gettext('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'
```

Ayrıca bakınız:

[Translation topic in the Django documentation](#)

3.7.5 Simgeler

Weblate currently uses material design icons. In case you are looking for new symbol, check [Material Design Icons](#) or [Material Design Resources](#).

Additionally, there is `scripts/optimize-svg` to reduce size of the SVG as most of the icons are embedded inside the HTML to allow styling of the paths.

3.8 Weblate sorunlarını bildirmek

Weblate [hata izleyicisi](#) GitHub üzerinde bulunur.

Feel welcome to report any issues you have, or suggest improvement for Weblate there. There are various templates prepared to comfortably guide you through the issue report.

If what you have found is a security issue in Weblate, please consult the [Güvenlik sorunları](#) section below.

If you are not sure about your bug report or feature request, you can try [Weblate tartışmaları](#).

3.8.1 Güvenlik sorunları

In order to give the community time to respond and upgrade, you are strongly urged to report all security issues privately. HackerOne is used to handle security issues, and can be reported directly at [HackerOne](#). Once you submit it there, community has limited but enough time to solve the incident.

Alternatively, report to security@weblate.org, which ends up on HackerOne as well.

If you don't want to use HackerOne, for whatever reason, you can send the report by e-mail to michal@weblate.org. You can choose to encrypt it using this PGP key `3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D`. You can also get the PGP key from [Keybase](#).

Not: Weblate depends on third-party components for many things. In case you find a vulnerability affecting one of those components in general, please report it directly to the respective project.

Bunların bazıları şunlardır:

- [Django](#)
 - [Django REST çatısı](#)
 - [Python Social Auth](#)
-

3.9 Weblate testsuite and continuous 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
```

İpucu: 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 [Weblate için veri tabanı kurulumu](#) 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
```

İpucu: The tests can also be executed inside developer docker container, see [Weblate platformunu Docker üzerinde yerel olarak çalıştırma](#).

Ayrıca bakınız:

See [Testing in Django](#) for more info on running and writing tests for Django.

3.10 Data schemas

Weblate uses [JSON Schema](#) to define layout of external JSON files.

3.10.1 Weblate çeviri belleği şeması

https://weblate.org/schemas/weblate-memory.schema.json		
tür	<i>dizi</i>	
öğeler	<i>Çeviri belleği ögesi</i>	
	tür	<i>nesne</i>
	özellikler	
	• category	<i>Dizge kategorisi</i>
		1 genel, 2 paylaşılmış, 10000000+ projeye özel, 200000000+ kullanıcıya özel
		tür <i>integer</i>
	examples	1
	minimum	0
	default	1
	• origin	Dizge kaynağı
		Dosya adı veya bileşen adı

[sonraki sayfaya devam](#)

Tablo 1 – önceki sayfadan devam

	tür	<i>string</i>
	examples	test.tmx project/component
• source	default	
	<i>Kaynak dizge</i>	
	tür	<i>string</i>
	examples	Merhaba
• source_language	minLength	1
	default	
	<i>Kaynak dil</i>	
	ISO 639-1 / ISO 639-2 / IETF BCP 47	
• target	tür	<i>string</i>
	examples	en
	pattern	^[^]+\$
	default	
• target_language	<i>Hedef dizge</i>	
	tür	<i>string</i>
	examples	Ahoy
	minLength	1
• target_language	default	
	<i>Hedef dil</i>	
	ISO 639-1 / ISO 639-2 / IETF BCP 47	
	tür	<i>string</i>
	examples	tr
	pattern	^[^]+\$
	default	
	additionalProperties	False
definitions		

Ayrıca bakınız:Çeviri belleği, *dump_memory*, *import_memory***3.10.2 Weblate kullanıcı verilerini dışa aktarma**

https://weblate.org/schemas/weblate-userdata.schema.json		
tür	<i>nesne</i>	
özellikler		
• basic	<i>Temel</i>	
	tür	<i>nesne</i>
	özellikler	
• username	<i>Kullanıcı adı</i>	
	tür	<i>string</i>
	examples	admin
	default	
• full_name	<i>Ad soyad</i>	
	tür	<i>string</i>
	examples	Weblate Yöneticisi
	default	
• email	<i>E-posta</i>	
	tür	<i>string</i>
	examples	noreply@ornek.com
	default	
	format	email

sonraki sayfaya devam

Tablo 2 – önceki sayfadan devam

• profile	• date_joined	Katılma tarihi	
		tür	string
		examples	2019-11-18T18:53:54.862Z
		default	
	additionalProperties	format	date-time
		False	
		Profil	
		tür	nesne
	• language	özellikler	
		Dil	
		tür	string
		examples	tr
	• suggested	pattern	^[^]*\$
		default	
		Önerilmiş dizge sayısı	
		tür	integer
	• translated	examples	1
		default	0
		Çevrilmiş dizge sayısı	
		tür	integer
	• uploaded	examples	24
		default	0
		Yüklenmiş ekran görüntüsü sayısı	
		tür	integer
	• hi-de_completed	examples	1
		default	0
		Tamamlanmış çevirileri panoda gizler	
		tür	boolean
	• secondary_in_zen	examples	False
		default	True
		Zen kipinde ikincil çevirileri görüntüler	
		tür	boolean
	• hi-de_source_sec	examples	True
		default	True
		İkincil çeviri varsa kaynağı gizler	
		tür	boolean
	• editor_link	examples	False
		default	True
		Düzenleyici bağlantısı	
		tür	string
	• translate_mode	examples	
		pattern	^.*\$
		default	
		Çeviri düzenleyici kipi	
	• zen_mode	tür	integer
		examples	0
		default	0
		Zen düzenleyici kipi	
	• special_chars	tür	integer
		examples	0
		default	0
		Özel karakterler	
		tür	string
		examples	
		pattern	^.*\$

sonraki sayfaya devam

Tablo 2 – önceki sayfadan devam

• auditlog	• dashboard_view	default	
		<i>Varsayılan pano görünümü</i>	
		tür	<i>integer</i>
		examples	1
		default	0
	• dashboard_components	<i>Varsayılan bileşen listesi</i>	
		default	null
		anyOf	tür <i>null</i>
			tür <i>integer</i>
	• languages	<i>Çevrilmiş diller</i>	
		tür	<i>dizi</i>
		default	
		öğeler	<i>Dil kodu</i>
		tür	<i>string</i>
		examples	tr
		pattern	^.*\$
		default	
	• secondary_languages	<i>İkincil diller</i>	
		tür	<i>dizi</i>
		default	
		öğeler	<i>Dil kodu</i>
		tür	<i>string</i>
		examples	sk
		pattern	^.*\$
		default	
	• watched	<i>İzlenen projeler</i>	
		tür	<i>dizi</i>
		default	
		öğeler	<i>Proje kısaltması</i>
		tür	<i>string</i>
		examples	weblate
		pattern	^.*\$
		default	
	• auditlog	additionalProperties	False
		<i>Denetim günlüğü</i>	
		tür	<i>dizi</i>
		default	
		öğeler	<i>Öğeler</i>
		tür	<i>nesne</i>
		özellikler	
		• address	<i>IP adresi</i>
		tür	<i>string</i>
		examples	127.0.0.1
		pattern	^.*\$
		default	
		• user_agent	<i>Kullanıcı uygulaması</i>
		tür	<i>string</i>
		examples	PC / Linux / Firefox 70.0
		pattern	^.*\$
		default	
		• timestamp	<i>Zaman damgası</i>
		tür	<i>string</i>
		examples	2019-11- 18T18:58:30.845Z

sonraki sayfaya devam

Tablo 2 – önceki sayfadan devam

• activity	default	
	format	date-time
	Etkinlik	
	tür	string
	examples	oturum açma
	pattern	^.*\$
	default	
additionalProperties	False	
definitions		

Ayrıca bakınız:*Kullanıcı profili, dumpuserdata*

3.11 Releasing Weblate

3.11.1 Releasing schedule

Weblate has two month release cycle for releases (x.y). These are usually followed by a bunch of bugfix releases to fix issues which slip into them (x.y.z).

The change in the major version indicates that the upgrade process can not skip this version - you always have to upgrade to x.0 before upgrading to higher x.y releases.

Ayrıca bakınız:*Upgrading Weblate*

3.11.2 Release planning

The features for upcoming releases are collected using GitHub milestones, you can see our roadmap at <<https://github.com/WeblateOrg/weblate/milestones>>.

3.11.3 Release process

Things to check prior to release:

1. Check newly translated languages by `./scripts/list-translated-languages`.
2. Set final version by `./scripts/prepare-release`.
3. Make sure screenshots are up to date `make -j 12 -C docs update-screenshots`.
4. Merge any possibly pending translations `wlc push; git remote update; git merge origin/weblate`

Perform the release:

5. Create a release `./scripts/create-release --tag` (see below for requirements).

Post release manual steps:

6. Update Docker image.
7. Close GitHub milestone.
8. Once the Docker image is tested, add a tag and push it.
9. Update Helm chart to new version.

10. Include new version in `.github/workflows/migrations.yml` to cover it in migration testing.
11. Increase version in the website download links.
12. Increase version in the repository by `./scripts/set-version`.
13. Check that readthedocs.org did build all translations of the documentation using `./scripts/rtd-projects`.

To create tags using the `./scripts/create-release` script you will need following:

- GnuPG with private key used to sign the release
- Push access to Weblate git repositories (it pushes tags)
- Configured **hub** tool and access to create releases on the Weblate repo
- SSH access to Weblate download server (the Website downloads are copied there)

3.12 Güvenlik ve gizlilik

Tüyo: Weblate üzerinde güvenlik, kullanıcılarımızın gizliliğine önem veren bir ortam sağlar.

Weblate geliştirme çalışmaları, [Linux Vakfı Temel Altyapı Girişiminin İyi Örneklerini](#) izlemektedir.

Ayrıca bakınız:

Güvenlik sorunları

3.12.1 Security updates

Only the latest release is guaranteed to receive security updates.

3.12.2 Güvenlik açıkları için bağımlılıkları izlemek

Bağımlılıklarımızdaki güvenlik sorunları [Dependabot](#) ile izlenir. Python ve JavaScript kitaplıkları kapsanır ve son kararlı sürümde, güvenlik açıklarını önlemek için bağımlılıklar güncellenir.

İpucu: Üçüncü taraf kitaplıklarında Weblate üzerinde etkisi olmayan güvenlik açıkları olabilir. Bu nedenle Weblate hata düzeltme sürümleri yayınlanmaz ve bunlar giderilmez.

3.12.3 Docker kapsayıcısı güvenliği

Docker kapsayıcıları [Anchore](#) ve [Trivy](#) güvenlik tarayıcıları ile düzenli olarak taranır.

Böylece, güvenlik açıklarını erken bulmamız ve iyileştirmeleri hızlı bir şekilde yayınlamamız sağlanır.

Bu taramaların sonuçlarını GitHub üzerinden alabilirsiniz. Bunlar CI üzerinde SARIF (Statik Analiz Sonuçları Değişim Biçimi) yapıları biçiminde tutulur.

Ayrıca bakınız:

Continuous integration

3.13 Weblate modüllerine katkıda bulunmak

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 istemcisi](#)
- `translation-finder`, used to discover translatable files in the repository
- `language-data`, language definitions for Weblate, see [Dil tanımları](#)

3.13.1 Kodlama rehberi

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
```

Ayrıca bakınız:

The CI integration is very similar to [Weblate testsuite and continuous 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
```

Ayrıca bakınız:

[Weblate kaynak kodları](#)

3.14 Weblate hakkında

3.14.1 Proje hedefleri

Çevirmenlerin katkıda bulunmasını kolaylaştıran, geniş bir *Desteklenen dosya biçimleri* yelpazesini destekleyen sıkı *Sürüm denetimi bütünleştirmesi* bütünleşmesi ile web temelli sürekli yerelleştirme aracı.

3.14.2 Proje adı

“Weblate”, “web” ve “translate” sözcüklerinin birleşimidir.

3.14.3 Proje web sitesi

Açılış sayfası <https://weblate.org> ve bulut barındırma hizmeti <https://hosted.weblate.org> adresinde bulunur. Belgeler <https://docs.weblate.org> adresinde görülebilir.

3.14.4 Proje logoları

Proje logoları ve diğer çizelgeler <https://github.com/WeblateOrg/graphics> adresinde bulunabilir.

3.14.5 Liderlik

Bu proje, michal@cihar.com adresinden ulaşılabilen Michal Čihař tarafından geliştiriliyor.

3.14.6 Katkıda bulunanlar

Weblate, Michal Čihař tarafından kuruldu. 2012 yılında kurulduğundan bu yana binlerce kişi katkıda bulundu.

3.15 Lisans

More detailed licensing information is available in the Weblate source code and follows [REUSE 3.0 specification](#).

Telif Hakkı © Michal Čihař michal@cihar.com

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/>.

4.1 Weblate 4.16

1 Mart 2023 tarihinde yayınlandı.

- Biçim dizesi denetimlerinin yinelenen biçimleri de algılaması sağlandı.
- Bazı özel biçimlendirilmiş dizgelerin arama başarımı iyileştirildi.
- Celery görev zamanlamasının kayıt yeri veri tabanı olarak değiştirildi.
- IBM Watson Language Translator desteği eklendi.
- 4.14 sürümünde kullanımdan kaldırılan sürüm denetimi sistemi bütünleştirmesi desteği kaldırıldı.
- Bitbucket sunucusu çekme istekleri desteği eklendi.
- Gettext po dosyalarında çakışmaların işlenmesi iyileştirildi.
- API ile ekleme sırasında dizge durumunu tanımlama desteği eklendi.
- CORS tarafından izin verilen kaynakları yapılandırma desteği eklendi.
- Çoğul biçimler için kendiliğinden öneri desteği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.2 Weblate 4.15.2

25 Ocak 2023 tarihinde yayınlandı.

- Varsayılan yapılandırmada gotext JSON ve i18next v4 biçimleri etkinleştirildi.
- Bozuk dosyalar yüklendiğinde yaşanan çökme sorunu düzeltildi.
- Git depo durumunda eski klasörlerin görüntülenmesi sağlandı.

Ayrıntılı olarak tüm değişiklikler.

4.3 Weblate 4.15.1

19 Ocak 2023 tarihinde yayınlandı.

- Kendiliğinden çeviri önerileri düzeltildi.
- Bazı uç durumlarda yaşanan eklenti sayfasının çökmesi sorunu düzeltildi.
- Bazı durumlarda yeni çevirilerde yaşanan kalıbın çevrilmemesi sorunu düzeltildi.
- **REUSE 3.0** kullanılarak lisanslama belgelendirildi.
- Takım yönetiminde kullanıcıların sayfalandırması düzeltildi.
- Proje oluşturma ve kaydetme başarımı iyileştirildi.
- gotext JSON dosyaları desteği eklendi.
- i18next v4 dosyaları desteği eklendi.
- API üzerinde sayfalandırma artık özelleştirilebilir.

Ayrıntılı olarak tüm değişiklikler.

4.4 Weblate 4.15

16 Aralık 2022 tarihinde yayınlandı.

- Tek bir dizge için değişikliklere göz atma özelliği eklendi.
- Diğer bileşenlerden kendiliğinden çeviri yapılırken çoğulların işlenmesi düzeltildi.
- Dizgeyi öneri olarak göndermek için Alt+Enter klavye kısayolu eklendi.
- Fluent biçiminde yerleştirilebilir ögeler desteği eklendi.
- Çeviri belleğinin başarımı iyileştirildi.
- İyi bilinen kod barındırma hizmetleri için repoweb tarama bağlantılarının kendiliğinden oluşturulması sağlandı.
- Birkaç görünümün başarımı iyileştirildi.
- Çoğul biçimleri olan dizgelerin listelenmesi iyileştirildi.
- HTML head bölümüne markup kodu eklenmesi sağlandı.
- Eklentide, MO dosyaları oluşturulurken yalnızca çevrilen dosyaların katılması sağlandı.
- Kurallı ifade işaretlerinin işlenmesi düzeltildi.
- Çoğul biçimlerde yer belirtici denetimi davranışı iyileştirildi.
- Çeviri dosyalarının Google Play ile uyumlu biçimde adlandırılması özelliği eklendi.
- API üzerine etiket desteği eklendi.
- İşlemeler için bildirimlerden farklı e-posta seçme özelliği eklendi.
- Docker kalıbı artık hata ayıklama kipini varsayılan olarak etkinleştirmiyor.
- Sözlük terimleri sözlük bileşeni önceliğine göre sıralandı.
- Takım üyeleri ekleyebilen ya da kaldırabilen takım yöneticileri eklendi.
- Kullanıcıları silmeden önce bir açılır pencere onayı eklendi.
- XML çıktısını özelleştirmek için eklenti eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.5 Weblate 4.14.2

5 Kasım 2022 tarihinde yayınlandı.

- Çeviri belleğinden kayıtları kaldırma özelliği eklendi.
- Çift dil uyarısında çözümleme iyileştirildi.
- Art arda yinelenen sözcük denetiminin doğruluğu iyileştirildi.
- Çok sayıda bildirim gönderme ölçeklendirilmesi iyileştirildi.
- Alt yazı çevirisi için dizge durumunun işlenmesi iyileştirildi.
- VCS hizmeti API anahtarlarının _CREDENTIALS listesi yerine _TOKEN/_USERNAME yapılandırması ile güvenli olmayan yapılandırması kullanımdan kaldırıldı.
- Yüklenen bazı CSV dosyalarının işlenmesi düzeltildi.
- Fark görünümünde boşluk değişimlerinin işlenmesi iyileştirildi.
- Yönetim sayfalarına kendiliğinden öneri yönetimi bağlantısı eklendi.
- Geçmiş yorumların kaldırılması/çözümlemesi için izleme eklendi.
- İlişkili bileşenlerle proje yedeklerinin geri yüklenmesi düzeltildi.
- Hesap açılmadığında captcha girişi düzeltildi.
- DeepL dil desteği iyileştirildi.
- Kimliği doğrulanmış depolarla web kancaları uyumluluğu iyileştirildi.
- Python 3.11 desteği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.6 Weblate 4.14.1

15 Eylül 2022 tarihinde yayınlandı.

- Bazı durumlarda yaşanan proje yedeklerinin oluşturulması sorunu düzeltildi.
- Dosya yüklemede hata bildirimi iyileştirildi.
- Kimlik doğrulama yapılırken GitHub üzerinden e-postaları doğrulanmış tüm kullanıcıların alınması sağlandı.
- Sözlük terimlerinin bağlam ya da anahtarlarla eşleştirmesi engellendi.
- Dizge kaldırma bildirimleri eklendi.
- Sözlükte çevrilemeyen terimlerin yönetimi iyileştirildi.
- Takım yönetimi sayfasında takım üyelerinin sayısı listelendi.
- Grup yönetimi arayüzü eklendi.
- Onaylama kullanıldığında onaylama istatistiklerinin her zaman görüntülenmesi sağlandı.
- Birim API uygulamasına arama özelliği eklendi.
- Onaylama iş akışında, salt okunur dizgeler için ilerleme çubuğunun görüntülenme sorunu düzeltildi.
- Birmanca için noktalama denetimleri iyileştirildi.
- Ölçüm verilerinin çöp biriktirmesi düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.7 Weblate 4.14

22 Ağustos 2022 tarihinde yayınlandı.

- Geçmişteki eklenti değişikliklerinin izlenmesi sağlandı.
- Windows RC, HTML ve metin dosyalarından yaşanan çeviri işleme sorunu düzeltildi.
- Dil kodu biçemi yapılandırma seçenekleri genişletildi.
- Son CLDR sürümlerinde güncellenen çoğul biçimler desteği eklendi.
- Çok sayıda çeviri içeren bileşenler güncellenirken bellek kullanımı azaltıldı.
- SAP Translation Hub çeviri etki alanı desteği eklendi.
- Kaynak dizge konumlarında mutlak bağlantıların kullanılabilmesi sağlandı.
- Bazı ters vekil sunucuların arkasında çalışma iyileştirildi.
- API, çeviri belleğini kapsayacak şekilde genişletildi.
- Belge çevirisi iş akışı iyileştirildi.
- HTML ve metin dosyaları çevirisinin güvenilirliği artırıldı.
- Proje düzeyinde yedekleme özelliği eklendi.
- Çeviri belleği aramalarının başarımı ve bellek kullanımı iyileştirildi.

[Ayrıntılı olarak tüm değişiklikler.](#)

4.8 Weblate 4.13.1

1 Temmuz 2022 tarihinde yayınlandı.

- Geçmiş kayıtlarında yaşanan önerilerin izlenmesi sorunu düzeltildi.
- Cloudflare üzerinden yaşanan ters vekil sunucu bilgisini işleme sorunu düzeltildi.
- Ayırıştırma hatasının bir bileşenin çevrilmesini kitlemesi sağlandı.
- Keşif eklentisinde yaşanan ara dosyayı yapılandırma sorunu düzeltildi.
- Yer belirticilerde DeepL çeviri davranışı düzeltildi.
- API üzerinden dizgelerin çevrilmemesi sorunu düzeltildi.
- API üzerine bir gruptan kullanıcı kaldırma özelliği eklendi.
- Kullanıcı davet e-postalarında yaşanan denetim günlüğü sorunu düzeltildi.
- Java biçimlendirme dizgelerinde yaşanan im adları sorunu düzeltildi.

[Ayrıntılı olarak tüm değişiklikler.](#)

4.9 Weblate 4.13

15 Haziran 2022 tarihinde yayınlandı.

- Dil adlarını güncelleme davranışı değiştirildi.
- Proje listesine sayfalandırma eklendi.
- Yeni birimler oluşturma API uygulaması artık yeni oluşturulan birim hakkında bilgi veriyor.
- Bileşen keşfinde bir ara dilin yapılandırılması eklendi.

- CSV biçimlerine sabit kodlama çeşitleri eklendi.
- Temeldeki uygulamaya daha iyi uyması amacıyla bazı biçimler için bağlam ve konumun işlenmesi değiştirildi.
- ResourceDictionary biçimi desteği eklendi.
- Renk körleri için ilerleme çubuğu renkleri iyileştirildi.
- Dizge kaldırılırken yaşanan çeşitlerin temizlenmesi sorunu düzeltildi.
- Django 4.1 uyumluluğu sağlandı.
- XLIFF içinde kaçırılan XML öğelerini depolama özelliği eklendi.
- Yer belirtici denetim hatalarının biçimlendirilmesi iyileştirildi.
- /.well-known/change-password/accounts/password/ adresine yönlendirildi.
- Makine çevirisi hizmetleri artık her proje için ayrı olarak yapılandırılabilir.
- Yorumların çözümlenmesi için ayrı bir izin eklendi ve *Dizgeleri onaylama* rolüne verildi.
- Alternatif çevirileri CSV dosyasında depolama özelliği eklendi.
- Yer belirtici denetimi artık büyük/küçük harfe duyarsız olarak da yapılabilir.

Ayrıntılı olarak tüm değişiklikler.

4.10 Weblate 4.12.2

11 Mayıs 2022 tarihinde yayınlandı.

- Bazı bileşenlerde yaşanan proje çeviri belleğinin yeniden oluşturulması sorunu düzeltildi.
- Bileşenlerin çevrilmemiş dizgelere göre sıralanması sorunu düzeltildi.
- Yeni dil eklenirken yaşanan olası çeviri kaybı sorunu düzeltildi.
- Aktarım işlemleri yapılırken Weblate SSH anahtarının oluşturulduğundan emin olunması sağlandı.

Ayrıntılı olarak tüm değişiklikler.

4.11 Weblate 4.12.1

29 Nisan 2022 tarihinde yayınlandı.

- Çekme isteği ileti başlığı düzeltildi.
- Fluent biçiminde söz dizimi hatalarının işlenmesi iyileştirildi.
- Bildirim e-postalarında yaşanan avatar görüntülenme sorunu düzeltildi.
- İnternet üzerinden para kazanma özelliği eklendi.
- Çeviriler kaldırılırken yaşanan eski kaynak dizgelerinin kaldırılması sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.12 Weblate 4.12

20 Nisan 2022 tarihinde yayınlandı.

- *Nokta işareti farklı* için Amharca desteği eklendi.
- *Soru işareti farklı* için Birmanca desteği eklendi.
- *Sözde yerel oluşturma* eklentisinin seçenekleri genişletildi.
- Bir dizgedeki tüm kalite denetimlerini yok saymak için `ignore-all-checks` işareti eklendi.
- *Sözde yerel oluşturma* eklentisinin denetim hatalarını tetiklemesi engellendi.
- *Gitea çekme isteği* özelliği eklendi.
- *Dil kodu biçemi* için Linux biçemi dil kodu eklendi.
- Proje çeviri belleğini yeniden oluşturma özelliği eklendi.
- API bir dosyadan bileşenler oluşturma için iyileştirildi.
- Diğer çevirilere kopyala ve çoğalt düğmeleri eklendi.
- Birleştirme isteği iletisi bileşen düzeyinde yapılandırılabilir kılındı.
- XML etiketlerinde en fazla uzunluk kısıtlama davranışı iyileştirildi.
- Fluent dosyalarının ek yorumlarla yüklenmesi sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.13 Weblate 4.11.2

4 Mart 2022 tarihinde yayınlandı.

- Binary sürümdeki bozuk MO dosyaları sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.14 Weblate 4.11.1

4 Mart 2022 tarihinde yayınlandı.

- Git ve Mercurial ile yaşanan argümanların eksik temizlenmesi sorunu düzeltildi - CVE-2022-23915. Ayrıntılı bilgi almak için [GHSA-3872-f48p-pxqj](#) bölümüne bakabilirsiniz.
- CSV dosyalarından belirsiz dizgelerin yüklenmesi sorunu düzeltildi.
- API üzerinden takım oluşturma özelliği eklendi.
- Kullanıcı anma önerilerinin görüntülenmesi sorunu düzeltildi.
- Proje belirteçlerine erişim artık özelleştirilebilir.

Ayrıntılı olarak tüm değişiklikler.

4.15 Weblate 4.11

25 Şubat 2022 tarihinde yayınlandı.

- XSS - CVE-2022-24710 sorunları düzeltildi. Ayrıntılı bilgi almak için [GHSA-6j6p-9rf9-gc66](#) adresine bakabilirsiniz.
- API kullanarak eklenti kurulumu sorunu düzeltildi.
- *İşlem gereken dizgeler* adı *Tamamlanmamış dizgeler* olarak değiştirildi.
- *ICU MessageFormat söz dizimi* kaynaklı hatalı pozitif sorunu düzeltildi.
- Diğer kullanımlar listesinde kilit ve katılımcı anlaşmasının görüntülenmesi sağlandı.
- Eski dizgeler ya da eksik çoğullar bulunduran PO dosyalarının güncellenme sorunu düzeltildi.
- Gerrit ile bir araya toplama eklentisi uyumluluğu iyileştirildi.
- Kullanıcı dillerinin *Accept-Language* üst bilgisine göre kendiliğinden hazırlanması sağlandı.
- Dizge kaldırılırken sorunların işlenmesi iyileştirildi.
- Weblate için artık Django 3.2 gerekiyor.
- Proje kodu kimlik doğrulaması ile bazı yazma işlemleri düzeltildi.
- Depodaki dizgeler değiştiğinde dizge durumunun izlenmesi sorunu düzeltildi.
- Depodaki dizge değişikliklerinin izlenmesi sağlandı.
- Gezinmeyi iyileştirmek için çeviriler listesine sabitlenebilen başlık eklendi.
- *Java özellikleri* ile yaşanan çevrilmeyen dizgeler sorunu düzeltildi.
- ASCII olmayan dal adlarıyla Git işlemi sorunu düzeltildi.
- *Çeviriyi kaynak dizgeyle önceden doldurma* eklentisi eklendi.
- *Hızlı ileri almadan birleştir* *Birleştirme biçemi* özelliği eklendi.
- Yeni eklenen dizgelerde *Kendiliğinden çeviri* eklentisinin tetiklenmesi sorunu düzeltildi.
- Birmanca için noktalama denetimleri iyileştirildi.
- Kullanıcılara erişim izni vermek için proje düzeyinde özel takımların tanımlanması sağlandı. Bilgi almak için: *Her proje için erişim denetimi yönetimi*.
- Uyarılara belge bağlantıları eklendi.
- Docker kapsayıcının, gerektiğinde giden e-posta için TLS/SSL seçeneğini kendiliğinden etkinleştirmesi sağlandı.
- Çözümlemiş yorumları arama özelliği eklendi.
- borgbackup 1.2 desteği eklendi.
- *Kendiliğinden çevrilmiş* etiketinin uygulanması sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.16 Weblate 4.10.1

22 Aralık 2021 tarihinde yayınlandı.

- Django 4.0 yükseltmesi ile gelen değişiklikler belgelendirildi.
- *Kendiliğinden çevrilmiş* etiketinin görüntülenmesi sorunu düzeltildi.
- Paylaşılan bir deposu olan bileşenlerde yaşanan dalın API ile görüntülenmesi sorunu düzeltildi.
- Başarısız olan itme uyarısının çözümüyle iyileştirildi.
- Değişikliklere göz atarken sayfayı el ile düzenleme sorunu düzeltildi.
- *Kashida harfi kullanılmış* doğruluğu iyileştirildi.
- Weblate Docker kapsayıcısı artık Python 3.10 kullanıyor.

Ayrıntılı olarak tüm değişiklikler.

4.17 Weblate 4.10

16 Aralık 2021 tarihinde yayınlandı.

- DeepL ile düzenlilik ve yer belirtici özelliği eklendi.
- Artık proje ve dil düzeyinde toplu düzenleme ve arama/değiştirme kullanılabilir.
- Arama ve değiştirme için süzme eklendi.
- Düzeltme: “Kendiliğinden çevir” yetkisi artık *Diller* grubunun bir parçası değil.
- “Kendiliğinden çevir”, *Yönetim* ve yeni *Kendiliğinden çeviri* grubuna eklendi.
- Özel karakterlerle XLSX dosyalarının oluşturulması sorunu düzeltildi.
- GitHub kimlik doğrulama yönetim bölümüne, kullanıcının belirli bir GitHub kuruluşunda veya takımında olup olmadığını denetleme özelliği eklendi.
- API üzerine iletilen geçersiz parametrelerle ilgili geri bildirim iyileştirildi.
- API için proje kapsamında erişim kodları özelliği eklendi.
- Bazı durumlarda yaşanan dizge silme sorunu düzeltildi.
- Yeni eklenen dizgelerin çevrilmesi sorunu düzeltildi.
- Süzülmesini kolaylaştırmak için kendiliğinden çevrilmiş dizgeler etiketlendi.

Ayrıntılı olarak tüm değişiklikler.

4.18 Weblate 4.9.1

19 Kasım 2021 tarihinde yayınlandı.

- Kalıp değiştirildikten sonra yaşanan tek dilli dosyaların yüklenmesi sorunu düzeltildi.
- İmlerdeki boşlukların işlenmesi iyileştirildi.
- İndirme API uygulamasına süzme özelliği eklendi.
- Yeni çeviriler eklerken yaşanan istatistik görünümü sorunu düzeltildi.
- GitHub SSH anahtar değişikliği ile ilgili sorunlar çözüldü.

Ayrıntılı olarak tüm değişiklikler.

4.19 Weblate 4.9

10 Kasım 2021 tarihinde yayınlandı.

- Geçmiş etkinliklerin ayrıntı düzeyi artırıldı.
- Geçmişin görüntülenmesi iyileştirildi.
- Çeviri sayfalarının başarımı iyileştirildi.
- Çeviri dosyası indirmelerini kısıtlama özelliği eklendi.
- `safe-html` artık `md-text` ile kullanıldığında Markdown kodlarını anlayabiliyor.
- `max-length` kod imi artık `xml-text` ile kullanıldığında XML kodlamasını yok sayıyor.
- *Çeviri en fazla uzunluğu* içinde oluşturulan metinlerin boyutları sorunu düzeltildi.
- Yaklaşan Google ilke değişikliklerine yardımcı olmak için uygulama mağazası başlığının uzunluğu 30 karaktere düşürüldü.
- SSH çağrısını `SSH_EXTRA_ARGS` aracılığıyla özelleştirme özelliği eklendi.
- ICU MessageFormat için denetimler eklendi.
- Makine çevirisi arka uçlarında hata durumunun ele alınması iyileştirildi.
- Dizgelerdeki olağan dışı boşluk karakterleri vurgulandı.
- Düzenleme yaparken çevrilen dizgede kalma seçeneği eklendi.
- Borg çağrısını `BORG_EXTRA_ARGS` aracılığıyla özelleştirme özelliği eklendi.
- Tek dilli çevirilerde yaşanan MO dosyalarının oluşturulması sorunu düzeltildi.
- Tüm bileşen çevirilerini ZIP dosyası olarak indirmek için API uç noktası eklendi.
- Python 3.10 desteği eklendi.
- Yönetim arayüzüne e-posta davetini yeniden gönderme özelliği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.20 Weblate 4.8.1

10 Eylül 2021 tarihinde yayınlandı.

- Django yönetici arayüzünde yaşanan kullanıcı silme sorunu düzeltildi.
- Eklenti parametreleri daha ayrıntılı olarak belgelendirildi.
- Sözlükte yaşanan JavaScript sorunu düzeltildi.
- Tutarlılık denetimlerinde eşleşme sayısına sınır eklendi.
- Makine çevirilerinde yer belirticilerin işlenmesi iyileştirildi.
- API kullanarak eklenti oluşturma sorunu düzeltildi.
- Alt bilgiye gizlilik ilkesi bağlantısı eklemek için `PRIVACY_URL` seçeneği eklendi.
- Üye e-posta adresleri proje yöneticilerinden gizlendi.
- Çakışma durumunda gettext PO birleştirmesi iyileştirildi.
- Sözlük vurgulaması iyileştirildi.
- XML denetimleriyle `safe-html` iminin davranışı iyileştirildi.
- Bağlantılı bileşenler için işleme iletileri düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.21 Weblate 4.8

21 Ağustos 2021 tarihinde yayınlandı.

- Apple stringsdict biçimine destek eklendi.
- PostgreSQL ile tam arama işlemine büyük/küçük harfe duyarlılık özelliği eklendi.
- Bazı durumlarda yaşanan sözlük açıklamalarının kaydedilmesi sorunu düzeltildi.
- Belgelendirmede iyileştirmeler yapıldı.
- Başarım iyileştirmeleri yapıldı.
- Gerrit ile bir araya toplama eklentisi uyumluluğu iyileştirildi.
- Tek dilli sözlük bileşenlerine dizge ekleme sorunu düzeltildi.
- Çeşitler işlenirken başarım iyileştirildi.
- Bir araya toplama eklentisinde yaşanan yukarı akış değişikliklerinin işlenmesini atlama sorunu düzeltildi.
- İndirmelerin dosya uzantısını korunsun.
- Fluent biçimi desteği eklendi.
- JSON biçimlerini girintilemek için sekmeleri kullanma özelliği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.22 Weblate 4.7.2

15 Temmuz 2021 tarihinde yayınlandı.

- Bir projede daha fazla dil kısaltmasının yapılandırılması özelliği eklendi.
- API üzerinde yaşanan arama dizgesini doğrulama sorunu düzeltildi.
- Bir etki alanı değişikliğinden sonra Git dışı aktarıcı adresleri düzeltildi.
- Windows RC dosyaları için temizleme eklentisi düzeltildi.
- XLIFF güncellemesinde yaşanabilecek çökme sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.23 Weblate 4.7.1

30 Haziran 2021 tarihinde yayınlandı.

- Sözlüğe terim ekleme açılır penceresi iyileştirildi.
- LibreTranslate makine çevirisi hizmeti desteği eklendi.
- Yeni projeler oluşturulurken hız sınırlaması eklendi.
- Dosya güncelleme başarımı iyileştirildi.

Ayrıntılı olarak tüm değişiklikler.

4.24 Weblate 4.7

17 Haziran 2021 tarihinde yayınlandı.

- Yapılandırma sağlığı denetimi iyileştirildi.
- gettext PO dosyasında kullanılan `object-pascal-format` desteği eklendi. Ayrıntılı bilgi: [Object Pascal biçimi](#).
- Amacı daha iyi açıklamak için *Yakındaki dizgeler*, *Benzer dizgeler* olarak yeniden adlandırıldı.
- *mi18n lang dosyaları* desteği eklendi.
- SAML kimlik doğrulama bütünleştirmesi iyileştirildi.
- Uç durumları daha iyi işlemek için *Gerrit* bütünleştirmesi düzeltildi.
- Weblate gereksinimi Django 3.2 olarak güncellendi.
- E-posta kimlik doğrulaması devre dışı bırakıldığında yaşanan kullanıcıları davet etme sorunu düzeltildi.
- Dil tanımları iyileştirildi.
- Kullanıcıların bir projeye katkıda bulunmalarını engelleme özelliği eklendi.
- Sözlük dillerinin kendiliğinden oluşturulması sorunu düzeltildi.
- Eklentilerle ilgili belgeler zenginleştirildi.
- Bağlantılı depoları olan bileşenlerin başarımları iyileştirildi.
- Ücretsiz DeepL API desteği eklendi.
- Kullanıcı yönetimi için artık Django yönetici arayüzüne gerek duyulmuyor.

Ayrıntılı olarak tüm değişiklikler.

4.25 Weblate 4.6.2

8 Mayıs 2021 tarihinde yayınlandı.

- Paylaşılan bileşenin projeler arasında taşınmasından sonra yaşanan çökme sorunu düzeltildi.
- Boş özellikler dosyalarına yeni dizgelerin eklenmesi sorunu düzeltildi.
- Sağdan sola yazılan dillerde kopya simgesinin hizalaması düzeltildi.
- Bilgi sekmesindeki dizge istatistikleri zenginleştirildi.
- Git üzerinde yok sayılan çeviri dosyalarının işlenmesi sorunu düzeltildi.
- Ölçüm başarımları iyileştirildi.
- Sözlükleri kaydederken ortaya çıkabilen sorun düzeltildi.
- Farklı çoğul kuralları olan dillerde tutarlılık denetiminin davranışı düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.26 Weblate 4.6.1

2 Mayıs 2021 tarihinde yayınlandı.

- Eski istenmeyen ileti koruma kodu kaldırıldı.
- Kaynak çoğul denetiminin doğruluğu iyileştirildi.
- Docker kullanıcı arayüzü dilleri listesi güncellendi.
- Çekme istekleri oluşturulurken alınan hata iletileri iyileştirildi.
- Pagure üzerinde çekme istekleri oluşturma sorunu düzeltildi.
- Kurulmuş eklentilerin kendiliğinden tetiklenmesi sorunu düzeltildi.
- Yükseltme yapılırken olası ön belleğe alma sorunları düzeltildi.
- Yükleme yapılırken tek dilli çevirilere yeni birimler eklenmesi sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.27 Weblate 4.6

19 Nisan 2021 tarihinde yayınlandı.

- auto_translate yönetim komutuna çeviri kipini belirten bir parametre eklendi.
- *Metin dosyaları* desteği eklendi.
- Tüm nesneler için eğilimler ve ölçümler eklendi.
- İkincil dillerdeki metinleri doğrudan kopyalama özelliği eklendi.
- Değişikliklere göz atarken tarihe göre süzme özelliği eklendi.
- Etkinlik çizelgeleri iyileştirildi.
- İletişim formu e-postalarının göndericisini yapılandırma özelliği eklendi.
- Bileşen oluşturma API uygulamasında parametre doğrulaması iyileştirildi.
- The rate limiting no longer applies to superusers.
- Kendiliğinden çeviri eklentisinin başarımı ve güvenilirliği iyileştirildi.
- Docker kapsayıcısına hız sınırlamasını özelleştirme özelliği eklendi.
- Bileşen oluşturma API uygulamasına kendiliğinden *İç Weblate adresleri* kullanma özelliği eklendi.
- Dizge listesinde durumun gösterilmesi sadeleştirildi.
- Parola karması varsayılan olarak Argon2 kullanacak biçimde güncellendi.
- Çeviri durumunu gösteren ilerleme çubukları sadeleştirildi.
- Amacı netleştirmek için *Eksik dilleri ekleme* adı değiştirildi.
- Dizge durumunu XLIFF olarak kaydetme sorunu düzeltildi.
- Dil genelinde arama özelliği eklendi.
- Docker dağıtımında ilk kez *Yatay ölçeklendirme* desteği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.28 Weblate 4.5.3

1 Nisan 2021 tarihinde yayınlandı.

- Ölçümlerin toplanması sorunu düzeltildi.
- Dizge eklenirken yaşanabilecek çökme sorunu düzeltildi.
- Arama sorgusu örnekleri iyileştirildi.
- Değiştirme yüklemesinde yaşanması olası yeni eklenen dizgelerin kaybolması sorunu düzeltildi.

4.29 Weblate 4.5.2

26 Mart 2021 tarihinde yayınlandı.

- Kendiliğinden çeviri için zamanlama yapılandırması eklendi.
- Lua biçim denetimi eklendi.
- *Art arda yinelenen sözcükler* denetiminde biçim dizgeleri yok sayıldı.
- Çeviri sayfasından ekran görüntüsü yüklenmesi sağlandı.
- Depo bakımına zorunlu dosya eşitlemesi eklendi.
- Daha uzun kodlu diller için kendiliğinden öneriler düzeltildi.
- Yeni dizge ekleme başarımı iyileştirildi.
- Kalite denetimlerinde çeşitli hata düzeltmeleri yapıldı.
- Çeşitli başarımlar iyileştirmeleri yapıldı.
- *Weblate uygulamasını keşfedin* ile bütünleştirme eklendi.
- Salt okunur dizgelerin denetlenmesi davranışı düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.30 Weblate 4.5.1

5 Mart 2021 tarihinde yayınlandı.

- Bazı uç durumlarda yaşanan sözlük imlerinin düzenlenmesi sorunu düzeltildi.
- Birkaç sayfanın başarımını artırmak için ölçümlerin kullanımı zenginleştirildi.
- TMX dosyalarında doğru kaynak dilin kaydedilmesi sağlandı.
- API kullanarak tek dilli PO yüklemelerinin işlenmesi iyileştirildi.
- Sözlük bileşenlerinde uyarı davranışı iyileştirildi.
- Markdown bağlantı denetimleri iyileştirildi.
- Öge yollarında sözlük ve kaynak dil belirtildi.
- Büyük projelerde bileşen listeleri sayfalandı.
- Çeviri, bileşen ya da proje silme başarımı iyileştirildi.
- Toplu düzenleme başarımı iyileştirildi.
- ODF dosyaları için “Düzenlenmesi gerekli” ve “Onaylandı” durumlarının korunması sorunu düzeltildi.
- Çeviri dosyası indirmelerini özelleştirme arayüzü iyileştirildi.

Ayrıntılı olarak tüm değişiklikler.

4.31 Weblate 4.5

19 Şubat 2021 tarihinde yayınlandı.

- gettext PO dosyasında kullanılan `lua-format` desteği eklendi.
- Bir bileşeni projeler arasında paylaşma özelliği eklendi.
- Birden çok biçim imiyle birden çok adsız değişken denetimi davranışı düzeltildi.
- Posta listesi alanı projelerden kaldırıldı ve yerine çevirmenler için genel yönergeler eklendi.
- Sahte yerel oluşturma eklentisi eklendi.
- TermBase eXchange dosyaları desteği eklendi.
- Bir im kullanarak dizge çeşitlerini el ile tanımlama desteği eklendi.
- Tutarlılık denetimlerinin başarımı iyileştirildi.
- Uzun dizgeler için çeviri belleği başarımı iyileştirildi.
- Açıklama metinlerinde arama desteği eklendi.
- İki dilli biçimlerde dizgelerin eklenmesi ve silinmesi özelliği eklendi.
- Amazon Translate makine çevirisinde desteklenen dillerin listesi zenginleştirildi.
- Java özellikleri için Java MessageFormat denetimleri kendiliğinden etkinleştirildi.
- Bir çeviriye yeni dizgeler eklemek için yeni bir yükleme yöntemi eklendi.
- Çeviriye göz atmak için basit bir arayüz eklendi.
- Sözlükler artık normal bileşenler olarak tutuluyor.
- Artık bileşen API uygulaması kullanıldığından sözlüklere özel API kaldırıldı.
- Bazı imleri değiştirmek için basitleştirilmiş arayüz eklendi.
- Sözlüğe çevrilemeyen veya yasak terimler desteği eklendi.
- Sözlüklere terminoloji tanımlama özelliği eklendi.
- Görsel klavyeye daha fazla alan sağlamak için metin yönünü değiştirme seçeneği taşındı.
- Kullanıcının katkıda bulunduğu projeleri kendiliğinden izleme seçeneği eklendi.
- Çevirinin sözlük ile eşleşip eşleşmediği denetimi eklendi.
- Gezinme metninin rengini belirleme özelliği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.32 Weblate 4.4.2

14 Ocak 2021 tarihinde yayınlandı.

- Dağıtılan bir MO dosyasının bozulması sorunu düzeltildi.

4.33 Weblate 4.4.1

13 Ocak 2021 tarihinde yayınlandı.

- Çoğul değişikliklerini geri alma sorunu düzeltildi.
- Proje ayarları için yardım görüntüleme sorunu düzeltildi.
- Kullanıcı yönetimi iyileştirildi.
- Tek dilli PO dosyalarında bağlamın işlenmesi iyileştirildi.
- Temizleme eklentisinin HTML, ODF, IDML ve Windows RC biçimlerine davranışı düzeltildi.
- CSV dosyalarında yaşanan konumun işlenmesi sorunu düzeltildi.
- Dosya indirmeleri için içerik sıkıştırmasının kullanılması sağlandı.
- ZIP dosyasından içe aktarmada kullanıcı deneyimi iyileştirildi.
- Yükleme dosya biçimlerinin algılanması iyileştirildi.
- Pagure için yinelenen çekme istekleri engellendi.
- Hayalet çevirilerin görüntülenmesi başarımı iyileştirildi.
- Çeviri düzenleyicisi yerel tarayıcı metin alanını kullanacak biçimde yeniden uygulandı.
- Temizleme eklentisinin eklenen yeni dizgeleri bozması sorunu düzeltildi.
- Eklentiler için API eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.34 Weblate 4.4

15 Aralık 2020 tarihinde yayınlandı.

- Bileşen oluştururken doğrulama iyileştirildi.
- Weblate gereksinimi Django 3.1 olarak güncellendi.
- Yönetim arayüzüne görünümü özelleştirme özelliği eklendi.
- Toplu düzenlemede yaşanan salt okunur durum işleme sorunu düzeltildi.
- CodeMirror bütünleştirilmesi iyileştirildi.
- Çeviri dosyalarından boş dizgeleri kaldırma eklentisi eklendi.
- Çeviriler için CodeMirror düzenleyicisinin kullanılması sağlandı.
- XML, HTML, Markdown ve reStructuredText için çeviri düzenleyiciye söz dizimi vurgulaması eklendi.
- Çeviri düzenleyicide yerleştirilebilir öğelerin vurgulanması sağlandı.
- Standart olmayan dil kodlarının desteği iyileştirildi.
- Belirsiz dil kodları kullanıldığında uyarı eklendi.
- Yeni bir çeviri eklenirken kullanıcıya süzölmüş bir dil listesinin sunulması sağlandı.
- Geçmişteki değişiklikler için arama yetenekleri zenginleştirildi.
- Fatura ayrıntı sayfaları ve Libre barındırma iş akışı iyileştirildi.
- Çeviri istatistikleri API uygulaması zenginleştirildi.
- Çeviri yaparken “diğer çeviriler” sekmesi iyileştirildi.
- Görevler API uygulaması eklendi.

- Dosya yükleme başarımı iyileştirildi.
- Kullanıcı tarafından tanımlanan özel karakterlerin görüntülenmesi iyileştirildi.
- Kendiliğinden çeviri başarımı iyileştirildi.
- Kullanıcı arayüzünde birkaç küçük iyileştirme yapıldı.
- ZIP indirmelerinin adlandırılması iyileştirildi.
- İzlenmeyen projelerin bildirimlerini alma seçeneği eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.35 Weblate 4.3.2

4 Kasım 2020 tarihinde yayınlandı.

- Belirli bileşen dosya maskeleriyle yaşanan çökme sorunu düzeltildi.
- Ard arda yinelenen sözcük denetiminin doğruluğu iyileştirildi.
- Pagure çekme istekleri desteği eklendi.
- Başarısız olan hesap açma işlemlerinin hata iletileri iyileştirildi.
- Geliştirici yorumlarının Markdown olarak görüntülenmesi geri alındı.
- “master” dalından farklı bir varsayılan dal kullanan Git depolarının kurulumu basitleştirildi.
- Yeni oluşturulan iç depoların varsayılan dal olarak main kullanması sağlandı.
- reStructuredText çevrilirken değiştirilmemiş çevirinin yanlış algılanma oranı azaltıldı.
- Bazı durumlarda karşılaşılan CodeMirror görüntüleme sorunları düzeltildi.
- Anlamını netleştirmek için Kalıp grubunun adı “Kaynaklar” olarak değiştirildi.
- Yolları uzun olan depolar için GitLab çekme istekleri düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.36 Weblate 4.3.1

21 Ekim 2020 tarihinde yayınlandı.

- Kendiliğinden çeviri başarımı iyileştirildi.
- Kimliği doğrulanmış kullanıcıların oturumlarının sona ermesi sorunu düzeltildi.
- Sürüm bilgilerini gizleme özelliği eklendi.
- Bitbucket sunucusuyla kanca uyumluluğu iyileştirildi.
- Çeviri belleği güncellemelerinin başarımı iyileştirildi.
- Bellek kullanımı azaltıldı.
- Matris görünümünün başarımı iyileştirildi.
- Bir kullanıcının bir projeden kaldırılması için onay adımı eklendi.

Ayrıntılı olarak tüm değişiklikler.

4.37 Weblate 4.3

15 Ekim 2020 tarihinde yayınlandı.

- API uygulamasına kullanıcı istatistikleri eklendi.
- Sayfalanmış sayfalardaki bileşen sıralaması sorunu düzeltildi.
- Bir sözlük için kaynak dil tanımlaması eklendi.
- GitHub ve GitLab çekme istekleri desteği yeniden yazıldı.
- Öneri kaldırıldıktan sonra istatistik sayılarının doğru olması sorunu düzeltildi.
- Herkese açık kullanıcı profili zenginleştirildi.
- Zorunlu denetimlerin yapılandırılması sorunu düzeltildi.
- İç yedekleme belgeleri iyileştirildi.
- Kaynak dil özniteliği projeden bir bileşene taşındı.
- Vue I18n biçimlendirme denetimi eklendi.
- Genel yer belirtici denetime kurallı ifade desteği eklendi.
- Matris kipinin görünümü iyileştirildi.
- Makine çevirisinin adı kendiliğinden öneriler olarak değiştirildi.
- Birden çok GitLab ya da GitHub kopyası ile etkileşim desteği eklendi.
- API, proje güncellemelerini, birim güncellemeleri ve silmeleri ile sözlükleri kapsayacak zenginleştirildi.
- Birim API uygulamasının çoğul dizgeleri düzgün şekilde işlemesi sağlandı.
- Bileşen oluşturma ZIP dosyası ya da belge yüklemesini işleyebilmesi sağlandı.
- API yanıt durum kodları birleştirildi.
- Katılımcı anlaşmasında Markdown kullanılabilmesi sağlandı.
- Kaynak dizge izlemesi iyileştirildi.
- JSON, YAML ve CSV biçimleriyle uyumluluk iyileştirildi.
- Dizgeleri silme özelliği eklendi.
- Dosya indirme başarımı iyileştirildi.
- Depo yönetimi görünümü iyileştirildi.
- Android için Java biçiminin kendiliğinden etkinleştirilmesi sağlandı.
- Yerelleştirilmiş ekran görüntüleri özelliği eklendi.
- Python 3.9 desteği eklendi.
- Belirli koşullarda yaşanan HTML dosyalarının çevrilmesi sorunu düzeltildi.

Ayrıntılı olarak tüm değişiklikler.

4.38 Weblate 4.2.2

2 Eylül 2020 tarihinde yayınlandı.

- JSON biçimlerinde yaşanan kaynak dizgelerin eşleşmesi sorunu düzeltildi.
- Bazı kimlik doğrulama yapılandırmaları için oturum açma yönlendirmesi düzeltildi.
- Grup eşitleme ile LDAP kimlik doğrulaması sorunu düzeltildi.
- Kendiliğinden çeviri ilerlemesini bildirirken yaşanan çökme sorunu düzeltildi.
- Art bilgiler etkinken Git işlemlerinin bir araya toplanması sorunu düzeltildi.
- API kullanarak yerel sürüm denetimi sistemi bileşenleri oluşturma sorunu düzeltildi.

4.39 Weblate 4.2.1

21 Ağustos 2020 tarihinde yayınlandı.

- Android kaynaklarında yaşanan bazı yerel ayarlar için çoğul biçimlerin kaydedilmesi sorunu düzeltildi.
- Bazı XLIFF dosyaları için temizleme eklentisinde yaşanan çökme sorunu düzeltildi.
- Docker kalıbında yerelleştirme CDN sunucusu kurulması sağlandı.

4.40 Weblate 4.2

18 Ağustos 2020 tarihinde yayınlandı.

- Kullanıcı sayfaları iyileştirildi ve kullanıcı listesi eklendi.
- 3.x sürümlerinden aktarım desteği kaldırıldı. 4.1 ya da 4.0 sürümünden aktarım yapın.
- Birkaç tek dilli biçime dışa aktarım özelliği eklendi.
- Etkinlik çizelgeleri iyileştirildi.
- Görüntülenecek Yakındaki dizgeler sayısının yapılandırılması sağlandı.
- Depo hataları ile karşılaşılan bileşenler için kilitleme desteği eklendi.
- Ana gezinme basitleştirildi (düğmeler simgelerle değiştirildi).
- Google Çeviri bütünleştirmesinde dil kodunun işlenmesi iyileştirildi.
- Git bir araya toplama eklentisi `Co-authored-by`: art bilgileri oluşturabilir.
- Arama sorgusu işleyici iyileştirildi.
- Biçim dizgesi denetimlerinin kullanıcıya bildirilmesi iyileştirildi.
- Toplu durum değişikliklerinin başarımlı iyileştirildi.
- Proje ya da bileşen yeniden adlandırıldıktan sonra uyumluluk yönlendirmeleri eklendi.
- Dizge onayı, bileşen kilitleme ve lisans değişikliği bildirimleri eklendi.
- ModernMT desteği eklendi.
- Dosya yüklenirken onaylanan çevirilerin üzerine yazılmasından kaçınılması sağlandı.
- Bazı uyumluluk adres yönlendirmelerinin desteği kaldırıldı.
- ECMAScript kalıp sabitleri denetimi eklendi.
- Bir bileşeni izleme seçeneği eklendi.

- JSON birim anahtarlarının başındaki nokta kaldırıldı.
- Çeviri belleği için ayrı Celery kuyruğu kaldırıldı.
- Tüm bileşenlerin bir dili bir kerede çevirmesi sağlandı.
- Content-Security-Policy HTTP üst bilgilerinin yapılandırılması sağlandı.
- Dillere proje düzeyinde takma ad verme özelliği eklendi.
- HTML ya da JavaScript yerelleştirmesine yardımcı olmak için yeni eklenti eklendi. Bilgi almak için: [JavaScript yerelleştirme CDN](#).
- Weblate etki alanının ayarlar bölümünden yapılandırılması sağlandı. Bilgi almak için: [SITE_DOMAIN](#).
- Bileşene ve projeye göre arama desteği eklendi.

4.41 Weblate 4.1.1

19 Haziran 2020 tarihinde yayınlandı.

- Docker üzerinde kendiliğinden düzeltme ya da eklenti yapılandırmasının değiştirilmesi sorunu düzeltildi.
- “Hakkında” sayfasında yaşanabilecek çökme sorunu düzeltildi.
- Bayt koduna derlenmiş yerel ayar dosyalarının kurulumu iyileştirildi.
- Sözlüğe sözcük ekleme sorunu düzeltildi.
- Makine çevirisi için klavye kısayolları sorunu düzeltildi.
- Bazı kurulumlarda günlük olaylarının atlanmasına neden olan hata ayıklama çıktısı kaldırıldı.
- Proje listesindeki kilit göstergesi sorunu düzeltildi.
- Bazı kurulumlarda GPG anahtarlarının listelenmesi sorunu düzeltildi.
- Kullanılacak DeepL API sürümünün seçme özelliği eklendi.
- SAML hizmet sağlayıcısı olarak davranma özelliği eklendi. Bilgi almak için: [SAML kimlik doğrulaması](#).

4.42 Weblate 4.1

15 Haziran 2020 tarihinde yayınlandı.

- Belirtilen ülke koduyla yeni çeviriler oluşturma özelliği eklendi.
- Ekran görüntüsü ile kaynak dizgeleri arama özelliği eklendi.
- İstatistik içgörülerindeki bilgiler zenginleştirildi.
- “Çeviri” sayfalarında arama düzenlemesi iyileştirildi.
- Eşzamanlı depo güncellemelerinin işlenmesi iyileştirildi.
- Proje oluşturma formuna kaynak dil eklendi.
- Kredilere değişiklik sayısı eklendi.
- Bazı durumlarda kullanıcı arayüzü dil seçimi düzeltildi.
- Hesap açma kapalıyken kayıt yöntemlerinin beyaz listeye eklenmesi sağlandı.
- Sözlükte ilgili terimlerin aranması iyileştirildi.
- Çeviri belleği eşleşmeleri iyileştirildi.
- Aynı makine sonuçları gruplandırıldı.
- Ekran görüntüsünü çeviri sayfasında düzenlemek için doğrudan bağlantı eklendi.

- Silme onayı penceresi iyileştirildi.
- ZIP indirmeye kalıplar katıldı.
- Duyurulara Markdown ve bildirim yapılandırması özellikleri eklendi.
- Denetim listelerindeki ayrıntılar zenginleştirildi.
- Yeni dosya biçimleri desteği eklendi: *Laravel PHP dizgeleri*, *HTML dosyaları*, *OpenDocument biçimi*, *IDML biçimi*, *Windows RC dosyaları*, *INI çevirileri*, *Inno Setup INI çevirileri*, *GWT özellikleri*, *go-i18n JSON dosyaları*, *ARB dosyası*.
- Yok sayılan denetimlerin durumu için tutarlı olarak yok sayıldı kullanıldı.
- Etkinleştirilecek varsayılan eklentileri yapılandırma özelliği eklendi.
- Yok sayılan denetimler için düzenleyici klavye kısayolu düzeltildi.
- Yer belirtici bulunan dizgelerin makine çevirisi iyileştirildi.
- Kullanıcı dillerini başlatmayı kolaylaştırmak için hayalet çevirinin görüntülenmesi sağlandı.
- Dil kodunun işlenmesi iyileştirildi.
- Kullanıcı dilindeki çevirilerin listenin ilk sırasında görüntülenmesi sağlandı.
- Şekillendiriciler adı daha genel olan Çeşitler olarak değiştirildi.
- Yeni kalite denetimleri eklendi: *Birden çok değişken adlandırılmamış*, *Uzun süredir çevrilmemiş*, *Art arda yinelenen sözcükler*.
- Çeviri belleğini boşaltma özelliği yeniden eklendi.
- Kaynak denetimlerini yok sayma seçeneği düzeltildi.
- Değişiklikleri itmek için farklı dallar yapılandırma özelliği eklendi.
- API uygulamasına HTTP üst bilgilerinde hız sınırlama durumunu bildirme özelliği eklendi.
- Google Translate V3 API (gelişmiş) desteği eklendi.
- Bileşen düzeyinde erişimi kısıtlama özelliği eklendi.
- Çeviri imlerinde boşluk ve diğer özel karakterleri kullanma desteği eklendi. Bilgi almak için: *Parametreleri kullanarak davranışı özelleştirmek*.
- Etkinse oluşturulan metin denetiminin her zaman görüntülenmesi sağlandı.
- API uygulamasına değişiklikleri süzme özelliği eklendi.
- Sözlüklerin projeler arasında paylaşılması özelliği eklendi.

4.43 Weblate 4.0.4

7 Mayıs 2020 tarihinde yayınlandı.

- Bazı Python 3.8 ortamlarında yaşanan deneme paketini çalıştırma sorunu düzeltildi.
- Belgelerdeki yazım hataları düzeltildi.
- Bazı durumlarda yaşanan API kullanılarak bileşenlerin oluşturulması sorunu düzeltildi.
- Mobil gezinmeyi bozan JavaScript hataları düzeltildi.
- Bazı denetimlerin görüntülenmesinde yaşanan çökme sorunu düzeltildi.
- Ekran görüntüleri listesi düzeltildi.
- Aylık özet bildirimleri düzeltildi.
- Çeviride bulunmayan birimlerdeki ara çeviri davranışı düzeltildi.

4.44 Weblate 4.0.3

2 Mayıs 2020 tarihinde yayınlandı.

- Raporlarda yaşanabilecek kilitlenme sorunu düzeltildi.
- Yorumlardaki kullanıcı anmalarının büyük/küçük harfe duyarsız olması sağlandı.
- Süper kullanıcı olmayan kullanıcılar için PostgreSQL aktarımı düzeltildi.
- Bileşen oluşturulurken depo adresinin değiştirilmesi sorunu düzeltildi.
- Yukarı akış deposu kaybolduğunda yaşanan kilitlenme sorunu düzeltildi.

4.45 Weblate 4.0.2

27 Nisan 2020 tarihinde yayınlandı.

- Çeviri istatistiklerinin başarımı iyileştirildi.
- Etiketleri değiştirme başarımı iyileştirildi.
- Toplu düzenleme başarımı iyileştirildi.
- Çeviri belleği başarımı iyileştirildi.
- Bileşen silinirken yaşanabilecek çökme sorunu düzeltildi.
- Bazı uç durumlarda yaşanan çeviri değişikliklerinin görüntülenme sorunu düzeltildi.
- Çok uzun Celery kuyruğu uyarısı iyileştirildi.
- Tutarlılık denetiminde yaşanması olası hatalı pozitif sorunu düzeltildi.
- Bağlı bileşen deposu değiştirilirken yaşanan kilitlenme sorunu düzeltildi.
- Değişiklik listesine, CSV dosyalarına ve raporlara düzenleme uzaklığı eklendi.
- Kanada Fransızcası için noktalama işareti aralığı denetimlerinde yaşanan hatalı pozitif sorunu düzeltildi.
- XLIFF dışı aktarmasında yaşanan yer belirteciler sorunu düzeltildi.
- Sıfır genişlik denetiminde yaşanan hatalı pozitif sorunu düzeltildi.
- Yapılandırma hatalarının raporlanması iyileştirildi.
- İki dilli kaynak yükleme sorunu düzeltildi.
- DeepL makine çevirisi için desteklenen dillerin kendiliğinden algılanması sağlandı.
- Bazı uç durumlarda yaşanan sabit ilerleme çubuğu görüntüsü sorunu düzeltildi.
- Çevrilmemiş dizgelerde yaşanan bazı denetimlerin tetiklenmesi sorunu düzeltildi.

4.46 Weblate 4.0.1

16 Nisan 2020 tarihinde yayınlandı.

- PyPI ile yaşanan paket kurulumu sorunu düzeltildi.

4.47 Weblate 4.0

16 Nisan 2020 tarihinde yayınlandı.

- Weblate için artık Python 3.6 ya da üzeri gerekiyor.
- Bileşen uyarılarına yönetim özeti eklendi.
- Bozuk depo tarayıcı adreslerine bileşen uyarısı eklendi.
- Oturum ve hesap açma sayfaları iyileştirildi.
- Proje erişim denetimi ve iş akışı yapılandırması proje ayarlarıyla bütünleştirildi.
- i18next için ara değerlendirme ve iç içe işleme denetimi ve vurgulayıcı eklendi.
- Yüzde yer belircileri için denetim ve vurgulayıcı eklendi.
- Denetim hatalarında önerilerin görüntülenmesi sağlandı.
- Geçmişe kaynak dizge değişikliklerinin kaydedilmesi sağlandı.
- Microsoft Translator, API 3 sürümüne yükseltildi.
- Çeviri belleği arka ucu yeniden yazıldı.
- *Arama* içine birkaç *is* : araması desteği eklendi.
- İç kara listeler için *Çeviri değiştirilmemiş* kullanılması sağlandı.
- Tek dilli po dosyalarındaki yorumların ayıklanması iyileştirildi.
- Beyaz tahta iletileri duyurular olarak yeniden adlandırıldı.
- Hesap açma e-postalarında bazen yaşanan sorunlar düzeltildi.
- Daha fazla sözdizimi çeşidini işlemek için LINGUAS güncelleme eklentisi iyileştirildi.
- Tek dilli XLIFF kaynak dosyasını düzenleme sorunu düzeltildi.
- *Arama* içine tam eşleşme desteği eklendi.
- API, ekran görüntüleri, kullanıcılar, gruplar, bileşen listeleri ve gelişmiş proje oluşturma işlemlerini kapsayacak biçimde zenginleştirildi.
- İki dilli çevirilere kaynak yükleme desteği eklendi.
- Geliştiricilerden ara dil desteği eklendi.
- Kaynak dizgeleri onaylama desteği eklendi.
- Platform genelinde çeviri belleği için indirme seçenekleri zenginleştirildi.

4.48 Weblate 3.x serisi

4.48.1 Weblate 3.11.3

11 Mart 2020 tarihinde yayınlandı.

- Belirli önceliği olan alanların aranması sorunu düzeltildi.
- Son eklenen dizgeler için önceden tanımlanmış sorgu düzeltildi.
- Aramada çift eşleşmelerin görüntülenmesi sorunu düzeltildi.
- Gmail üzerinde bildirimlerin görüntülenmesi sorunu düzeltildi.
- Geçmişteki değişikliklerin geri alınması sorunu düzeltildi.
- Özet bildirimlerindeki etkinliklere bağlantılar eklendi.

- Hesap silme onayı e-postası düzeltildi.
- Docker kapsayıcısında Slack kimlik doğrulaması desteği eklendi.
- Abone olunmayan diller için bildirim gönderilmesi engellendi.
- Başarım özetine Celery kuyrukları katıldı.
- Eklentiler için belge bağlantıları sorunu düzeltildi.
- Değiştirilmemiş çeviri denetiminde hatalı negatifler azaltıldı.
- CVE-2020-6802 sorununu çözmek için bleach bağımlılığı artırıldı.
- Geçmişte projesi düzeyindeki değişikliklerin listelenmesi sorunu düzeltildi.
- Bazı uç durumlarda yaşanan istatistiklerin geçersiz kılınması sorunu düzeltildi.
- Belirli dizge durumları için arama düzeltildi.
- Biçim dizgesi denetimi, eksik yüzde simgesini algılayacak şekilde iyileştirildi.
- Bazı üçüncü taraf hizmet sağlayıcılarında yaşanan kimlik doğrulama sorunu düzeltildi.

4.48.2 Weblate 3.11.2

22 Şubat 2020 tarihinde yayınlandı.

- Önerilerin görüntülenmesi düzeltildi.
- Bazı dizgelerin yanlış bir biçimde sözcük içermediğinin bildirilmesine yol açan sorun düzeltildi.

4.48.3 Weblate 3.11.1

20 Şubat 2020 tarihinde yayınlandı.

- Celery kurulum değişiklikleri belgesi yazıldı.
- Bileşen oluşturulurken dosya adı doğrulaması iyileştirildi.
- Bazı bağımlılıkların en düşük sürümleri düzeltildi.
- Belirli Django sürümlerini kullanan grupların eklenmesi sorunu düzeltildi.
- Yukarı akış deposuna el ile itme sorunu düzeltildi.
- Sözlük eşleştirmesi iyileştirildi.

4.48.4 Weblate 3.11

17 Şubat 2020 tarihinde yayınlandı.

- API ile bileşen oluşturulurken sürüm denetimi sistemi itme adresinin kullanılması sağlandı.
- Görüntülenen genişlik denetimi artık görüntüye görseli de ekliyor.
- Bildirim e-postalarındaki bağlantılar düzeltildi.
- Düz metin içerikli e-postaların görünümü iyileştirildi.
- Yok sayılan denetimlerin görüntülenmesi ve yeniden etkinleştirilmesi sağlandı.
- Tek dilli çevirilerde yakın anahtarların görüntülenmesi sağlandı.
- Dizge şekillendirmelerini gruplama özelliği eklendi.
- Sistem denetimlerine yeni Weblate sürümlerine yükseltme önerisi eklendi.
- Yinelenen dil uyarısı için daha ayrıntılı inceleme sağlandı.

- Proje sayfalarına ayrıntılı lisans bilgileri eklendi.
- Gerektiğinde yerel kopyaların sıklık düzeyinin kendiliğinden kaldırılması sağlandı.
- İşlem gerektiren dizgelerin indirilmesinde yaşanan sorun düzeltildi.
- Aynı dosya modelinin iki kez kullanılmasına karşı uyarı eklendi.
- Yerleştirilebilir XML öğelerinin ayıklanması iyileştirildi.
- `SINGLE_PROJECT` ayarına seçilmiş projeye yönlendirme dayatması eklendi.
- Yorumları çözümleme özelliği eklendi.
- İmleri toplu olarak düzenlenme özelliği eklendi.
- labels desteği eklendi.
- Toplu düzenleme eklentisi eklendi.
- *Denetimleri dayatmak* seçeneği eklendi.
- Onay bağlantılarının varsayılan geçerlilik süresi artırıldı.
- Matomo bütünleşmesi iyileştirildi.
- Kaynak dizge değişikliklerinin düzgün işlenmesi için *Çevrilmiş* düzeltildi.
- Kendiliğinden güncelleme yapılandırması `AUTO_UPDATE` seçeneği ile zenginleştirildi.
- LINGUAS eklentilerinin Weblate üzerindeki çevirileri tam olarak eşitlemesi sağlandı.

4.48.5 Weblate 3.10.3

18 Ocak 2020 tarihinde yayınlandı.

- translate-toolkit 2.5.0 desteği eklendi.

4.48.6 Weblate 3.10.2

18 Ocak 2020 tarihinde yayınlandı.

- Projelere kilit göstergesi eklendi.
- Bazı web tarayıcılarında titremeye neden olan CSS hatası düzeltildi.
- Yerel ayarları İngilizce olmayan sistemlerde yaşanan arama sorunu düzeltildi.
- GitHub ve Bitbucket kancaları için depo eşleştirmesi iyileştirildi.
- Bazı Python 2.7 kurulumlarında yaşanan veri aktarımı sorunları düzeltildi.
- Git için sıklık düzeyi ile kopyalama yapılandırılması özelliği eklendi.
- Arka planda bildirimlerin işlenmesi iyileştirildi.
- Web tarayıcısına geri dönerken yaşanan formun bozuk gönderilmesi sorunu düzeltildi.
- YAML biçimlendirmesini yapılandırmak için yeni eklenti eklendi.
- Tek çoğul biçimli dillerde tetiklenmemesi için çoğul biçimler aynı denetimi düzeltildi.
- Bazı alanlarda kurallı ifade aramasında yaşanan sorun düzeltildi.

4.48.7 Weblate 3.10.1

9 Ocak 2020 tarihinde yayınlandı.

- API çeviri oluşturma özelliği ile zenginleştirildi.
- Veri aktarımlarında yaşanan bazı uç durumlar düzeltildi.
- Django 3.0 uyumluluğu sağlandı.
- Veri temizleme başarımı iyileştirildi.
- Özelleştirilebilir security.txt desteği eklendi.
- Değişiklik günlüğünde sayfa yolları iyileştirildi.
- Panoda çeviri listesinin görünümü iyileştirildi.
- Web kancaları için HTTP yanıtları iyileştirildi.
- Docker kapsayıcısına GitLab birleştirme istekleri desteği eklendi.

4.48.8 Weblate 3.10

20 Aralık 2019 tarihinde yayınlandı.

- Uygulama kullanıcı arayüzü iyileştirildi.
- Çift boşluk denetimi eklendi.
- Yeni diller oluşturulurken yaşanan sorun düzeltildi.
- Silinmiş e-posta adreslerine denetim günlüğü bildirimlerinin gönderilmesi engellendi.
- Salt okunur dizge desteği eklendi.
- Yorumlara Markdown desteği eklendi.
- Proje bilgilerine çeviri yönerge metninin yerleştirilmesi sağlandı.
- İkincil diller için panoya kopyalama özelliği eklendi.
- Mercurial desteği iyileştirildi.
- Git deposundan alma başarımı iyileştirildi.
- Dizge yaşını arama özelliği eklendi.
- Tüm çeviriler için kaynak dilin görüntülenmesi sağlandı.
- Yakın dizgelerin bağlamının görüntülenmesi sağlandı.
- Depo işlemleriyle ilgili bildirimler desteği eklendi.
- Çevirilerin listelenmesi iyileştirildi.
- Arama yetenekleri zenginleştirildi.
- Düzenlenecek olarak imlenmiş kendiliğinden çeviri dizgeleri desteği eklendi.
- Bağlı bileşen uyarıları için yinelenen bildirimlerin gönderilmesi engellendi.
- Varsayılan birleştirme isteği iletisi iyileştirildi.
- Zen kipinde dizge durumunun daha iyi belirtilmesi sağlandı.
- Yandex Translate için daha fazla dil desteği eklendi.
- Bildirim e-postalarının görünümü iyileştirildi.
- Çeviri lisansı için seçenek sunuldu.

4.48.9 Weblate 3.9.1

28 Ekim 2019 tarihinde yayınlandı.

- Yedeklerden bazı gereksiz dosyalar kaldırıldı.
- Raporlardaki yaşanması olası kilitlenme sorunları düzeltildi.
- Veri tabanları arasında aktarma sorunu düzeltildi.
- Git depolarına dayatarak itme desteği eklendi.
- Kayıt belirtecinin geçersiz kılınması riski azaltıldı.
- Hesap silme işleminin hız sınırlayıcıya takılma sorunu düzeltildi.
- Önceliğe göre arama özelliği eklendi.
- JSON dosyasına dizge eklenirken yaşanabilecek çökme sorunu düzeltildi.
- Güvenli HTML denetimi ve düzeltmesinin kaynak dizgenin markup koduna uyması sağlandı.
- Davet edilmiş ve silinmiş kullanıcılara bildirim gönderilmesi engellendi.
- Docker kapsayıcısında Celery içindeki Redis SSL bağlantısı düzeltildi.

4.48.10 Weblate 3.9

15 Ekim 2019 tarihinde yayınlandı.

- İndirilen dosyalara Weblate üst verileri eklendi.
- Başarısız olan denetimler için kullanıcı arayüzü iyileştirildi.
- Biçim denetimlerinde eksik dizgeler belirtildi.
- Fransızca noktalama aralığı için özel denetim eklendi.
- Bazı kalite denetimi hatalarını düzeltme desteği eklendi.
- Yeni projeler oluşturmak için ayrı izinler eklendi.
- İstatistikler karakter sayıları ile zenginleştirildi.
- Java biçimi dil kodları desteği iyileştirildi.
- Yer belirteciler için yeni genel denetim eklendi.
- WebExtension JSON yer belirtecileri desteği eklendi.
- Düz XML biçimi desteği eklendi.
- API, proje, bileşen ve çeviri silme ve oluşturma özellikleri ile zenginleştirildi.
- Gitea ve Gitee web kancaları desteği eklendi.
- Yeni özel kurallı ifade temelli denetim eklendi.
- Paylaşılan çeviri belleğine katkıda bulunma yapılandırması sağlandı.
- Birden fazla çeviri dosyasını indirmek için ZIP desteği eklendi.
- XLIFF standardı, en fazla genişlik ve yazı tipi işlenmesi ile uyumlu hale getirildi.
- Web uygulamalarını çevirmek üzere güvenli HTML biçimlendirmesi için yeni denetim ve düzeltici eklendi.
- Desteklenmeyen yapılandırma için bileşen uyarısı eklendi.
- İlk çevirileri hazırlamak için kendiliğinden çeviri eklentisi eklendi.
- Kendiliğinden çeviri, önerileri eklemek üzere zenginleştirildi.
- Özet görünümüne eklenti parametreleri eklendi.

- Sentry için Raven yerine modern Sentry SDK desteği eklendi.
- Örnek ayarları, üretim ortamına daha uygun olacak biçimde değiştirildi.
- BorgBackup kullanılarak kendiliğinden yedekleme özelliği eklendi.
- İstenmeyen dosya güncellemelerini önlemek için RESX için temizleme eklentisi ayrıldı.
- Gelişmiş arama özellikleri eklendi.
- Kullanıcıların kendi raporlarını indirmeleri sağlandı.
- Bileşenlerin yapılandırılmasına yardımcı olacak yerelleştirme rehberi eklendi.
- GitLab birleştirme istekleri desteği eklendi.
- Depo durumunun görüntülenmesi iyileştirildi.
- Kendiliğinden çeviri işlemlerinin arka planda yapılması sağlandı.

4.48.11 Weblate 3.8

15 Ağustos 2019 tarihinde yayınlandı.

- Benzer bileşenlerin basitleştirilmiş oluşturulması desteği eklendi.
- XML temelli dosya biçimleri için çeviri imlerini işleme desteği eklendi.
- İstisnaların Celery günlüğüne kaydedilmesi sağlandı.
- Depo kapsamındaki eklentilerin başarımı iyileştirildi.
- Bildirim e-postalarının görünümü iyileştirildi.
- Parola sıfırlama davranışı düzeltildi.
- Çeviri sayfalarının çoğunda başarımlar iyileştirildi.
- Weblate tarafından bilinmeyen dillerin listesi düzeltildi.
- Keşfedilen bileşenlere eklentileri kopyalama özelliği eklendi.
- Dosya içeriğinin yüklenenlerle değiştirilmesi desteği eklendi.
- Sürüm denetimi sistemi temelli olmayan içeriğin çevrilmesi desteği eklendi.
- Sosyal ağlarda kullanılmak üzere OpenGraph pano bileşeni görseli eklendi.
- Canlandırılmalı ekran görüntüleri desteği eklendi.
- Tek dilli XLIFF dosyalarının işlenmesi iyileştirildi.
- Tek bir etkinlik için birden fazla bildirim gönderilmesi engellendi.
- Değişiklikleri süzme desteği eklendi.
- Raporlama için önceden tanımlanmış süreler zenginleştirildi.
- Azure depoları için web kancası desteği eklendi.
- Bekleyen öneriler veya çevrilmemiş dizgeler hakkında yeni abone olma bildirimleri eklendi.
- Bildirim e-postalarına tek tıkla abonelikten ayrılma bağlantısı eklendi.
- Çevrilmiş denetiminde yaşanan hatalı pozitif sorunu düzeltildi.
- Yöneticiler için yeni yönetim arayüzü eklendi.
- Dizge önceliğinin imler kullanılarak belirlenmesi sağlandı.
- Dil yönetimi görünümleri eklendi.
- Qt kitaplığı ve Ruby biçim dizgeleri için denetimler eklendi.
- Tek proje kurulumlarına daha uygun yapılandırma eklendi.

- Tek dilli çevirilerdeki kaynak dizge değişikliğinde yeni dizge için bildirim eklendi.
- Arama özelliği olan çeviri belleği için ayrı bir görünüm eklendi.

4.48.12 Weblate 3.7.1

28 Haziran 2019 tarihinde yayınlandı.

- Belgeler güncellendi.
- Bazı gereksinim kısıtlamaları düzeltildi.
- Dil veri tabanı güncellendi.
- Yerelleştirme güncelleştirmeleri yapıldı.
- Çeşitli kullanıcı arayüzü iyileştirmeleri yapıldı.
- Desteklenmeyen ancak keşfedilen çeviri dosyalarının işlenmesi iyileştirildi.
- Eksik dosya biçimi gereksinimlerinin daha ayrıntılı olarak bildirilmesi sağlandı.

4.48.13 Weblate 3.7

21 Haziran 2019 tarihinde yayınlandı.

- Bildirimler için ayrı Celery kuyruğu eklendi.
- API ile göz atarken uygulamayla tutarlı bir görünüm kullanıldı.
- Raporlara onaylanmış istatistikleri eklendi.
- Çeviri bileşeni güncellenirken ilerleme durumunun bildirilmesi sağlandı.
- Çalışan arka plan bileşeni güncellemesini iptal etme özelliği eklendi.
- Dosya adı değiştirme işlemleri için kalıp dili zenginleştirildi.
- Düzenleyici bağlantısı ve depo tarayıcı adresi için kalıplar kullanıldı.
- Çeviri düzenirken en fazla uzunluk ve geçerli karakter sayısının belirtilmesi sağlandı.
- Değiştirilmemiş çeviri denetiminde kısaltmaların işlenmesi iyileştirildi.
- Yeni katkıda bulunanlar için giriş sayfası yenilendi.
- msgmerge eklentisini yapılandırma desteği eklendi.
- Bildirim gönderirken SMTP bağlantısının açılması geciktirildi.
- Hata günlüğü iyileştirildi.
- MO oluşturma eklentisinde özel konum kullanılması sağlandı.
- Eski önerileri veya yorumları temizleme eklentileri eklendi.
- Zen düzenleyicisinde yatay kipi etkinleştirme seçeneği eklendi.
- Birçok bağlı bileşenle içe aktarma başarımı iyileştirildi.
- Bazı durumlardaki kurulum örnekleri düzeltildi.
- Değişiklik uyarılarının görüntülenmesi iyileştirildi.
- Yeni yatay istatistikler pano bileşeni eklendi.
- Çoğul biçimler için biçim dizgeleri denetimi iyileştirildi.
- Yazı tipi yönetim aracı eklendi.
- Görüntülenen metin boyutları için yeni denetim eklendi.

- Alt yazı biçimleri desteği eklendi.
- Diller için genel tamamlanma istatistikleri eklendi.
- Proje kapsamı ve genel kapsamda raporlama özelliği eklendi.
- Çeviri durumunu görüntüleyen kullanıcı arayüzü iyileştirildi.
- Weblate logosu ve renk şeması yenilendi.
- Bit eşlem simgelerinin görünümü yenilendi.

4.48.14 Weblate 3.6.1

26 Nisan 2019 tarihinde yayınlandı.

- Tek dilli XLIFF dosyalarının işlenmesi iyileştirildi.
- Bazı uç durumlarda yaşanan özet bildirimleri sorunu düzeltildi.
- Eklenti betiği hata uyarısı sorunu düzeltildi.
- Tek dilli PO dosyaları için MO dosyası oluşturma sorunu düzeltildi.
- Kaldırılan denetimlerin görüntülenmesi sorunu düzeltildi.
- Proje listesinde yönetilen projelerin belirtilmesi sağlandı.
- Eksik sürüm denetimi sistemi deposundan kurtarma için güncellemeye izin verildi.

4.48.15 Weblate 3.6

20 Nisan 2019 tarihinde yayınlandı.

- Kullanıcı verilerini indirme desteği eklendi.
- Eklentilerin kurulum sırasında kendiliğinden tetiklenmesi sağlandı.
- Aktarım çakışmalarını çözümlmek için yönergeler iyileştirildi.
- Temizleme eklentisinin uygulama mağazası üst veri çevirileriyle uyumlu olması sağlandı.
- Yeni çeviriler eklenirken dil kodu sözdiziminin yapılandırılabilmesi sağlandı.
- 2020 Nisan ayında desteğinin bitmesi planlanan Python 2 sürümünün kullanıldığı ile ilgili uyarı eklendi.
- Görsel klavye için kaynak dizgeden özel karakterlerin ayıklanması sağlandı.
- Katılımcı istatistikleri, kaynak ve hedef sayılarını yansıtacak şekilde zenginleştirildi.
- Çeviriler kullanıcılar için devre dışı bırakılmış olsa bile, yöneticilerin ve tutarlılık eklentilerinin çeviri yapabilmesi sağlandı.
- Language-Team üst bilgi değişikliğini devre dışı bırakma düğmesinin açıklaması düzeltildi.
- Yorumlarda anılan kullanıcıların bilgilendirilmesi sağlandı.
- Bileşen kurulumunda dosya biçiminin kendiliğinden algılanması özelliği kaldırıldı.
- Tek dilli PO dosyaları için MO dosyası oluşturma sorunu düzeltildi.
- Özet bildirimleri eklendi.
- Bileşen bildirimlerini sessize alma desteği eklendi.
- Yeni uyarılar, beyaz tahta iletileri ve bileşenler için bildirimler eklendi.
- Yönetilen projelerin bildirimlerinin yapılandırılabilmesi sağlandı.
- Üç harfli dil kodlarının işlenmesi iyileştirildi.

4.48.16 Weblate 3.5.1

10 Mart 2019 tarihinde yayınlandı.

- Celery systemd birimi örneği düzeltildi.
- Oturum açma ile HTTP depolarından gelen bildirimler düzeltildi.
- Tek dilli çeviriler için kaynak dizgeyi düzenlemedeki yarış durumu düzeltildi.
- Günlüklere başarısız olan eklenti çalıştırma çıktısı eklendi.
- Yeni dil ekleme seçeneklerinin doğrulanması iyileştirildi.
- Bileşen ayarlarında dosya biçimlerinin düzenlenmesi sağlandı.
- Kurulum yönergeleri Python 3 kullanacak biçimde güncellendi.
- Çevirilerin yüklenmesinde başarımlar ve tutarlılık iyileştirmeleri yapıldı.
- Microsoft Terminology hizmetinin geçerli Zeep sürümleriyle uyumlu olması sağlandı.
- Yerelleştirme güncelleştirmeleri yapıldı.

4.48.17 Weblate 3.5

3 Mart 2019 tarihinde yayınlandı.

- İç çeviri belleğinin başarımlarını iyileştirildi.
- Genel çeviri belleği yönetimi arayüzü eklendi.
- Hatalı bileşen durumu uyarısı iyileştirildi.
- Beyaz tahta iletilerini yönetme kullanıcı arayüzü eklendi.
- Eklenti işleme iletilerinin yapılandırılabilmesi sağlandı.
- Yukarı akış deposu güncellenirken işleme sayısı azaltıldı.
- Bileşenler projeler arasında taşınırken yaşanabilecek üst veri kaybı sorunu düzeltildi.
- Zen kipinde gezinme iyileştirildi.
- Birkaç yeni kalite denetimi eklendi (Markdown ve adresler ile ilgili).
- Uygulama mağazası üst veri dosyaları desteği eklendi.
- GitHub ya da Gerrit bütünleştirmesini değiştirme desteği eklendi.
- Kashida harfleri için denetim eklendi.
- İşlemleri sorumluya göre bir araya toplama seçeneği eklendi.
- XLSX dosya biçimi desteği iyileştirildi.
- Tesseract 4.0 ile uyumluluk sağlandı.
- Faturalama eklentisine, ödenmemiş faturalar için 45 gün sonra projeleri silme özelliği eklendi.

4.48.18 Weblate 3.4

22 Ocak 2019 tarihinde yayınlandı.

- XLIFF yer belirticileri desteği eklendi.
- Birden fazla Celery görev kuyruğu kullanma özelliği eklendi.
- Projeleri ve bileşenleri yeniden adlandırma ve taşıma desteği eklendi.
- Raporlara karakter sayıları eklendi.
- Çeviri dosyalarının kendiliğinden algılanmasıyla çeviri bileşenlerinin yönlendirme ile oluşturulması özelliği eklendi.
- Git işlemleri birleştirme iletilerinin özelleştirilebilmesi sağlandı.
- Gezinmede bileşen uyarılarının görsel olarak görüntülenmesini sağlandı.
- Çeviri dosyalarını yükleme başarımı iyileştirildi.
- Yollamadan önce Git işlemlerini bir araya toplama eklentisi eklendi.
- Çeviri değişikliklerinin görüntülenmesi iyileştirildi.
- Varsayılan birleştirme biçemi değiştirilerek yeniden temellendirildi ve yapılandırılabilmesi sağlandı.
- Dil kodundaki özel kullanım alt etiketlerinin işlenmesi iyileştirildi.
- Tam metin dizini güncelleme başarımı iyileştirildi.
- Dosya yükleme API işlevi daha fazla parametreyi destekleyecek biçimde zenginleştirildi.

4.48.19 Weblate 3.3

30 Kasım 2018 tarihinde yayınlandı.

- Bileşen ve proje silme desteği eklendi.
- Bazı tek dilli çeviriler için başarımlar iyileştirildi.
- Bir çeviriyle ilgili sorunları vurgulamak için çeviri bileşeni uyarıları eklendi.
- Yapılabildiğinde, XLIFF dizgesi resname değerinin bağlam olarak sunulması sağlandı.
- XLIFF durumları desteği eklendi.
- DATA_DIR içindeki yazılamayan dosyalar denetimi eklendi.
- Değişiklikler için CSV dışı aktarma işlemi iyileştirildi.

4.48.20 Weblate 3.2.2

20 Ekim 2018 tarihinde yayınlandı.

- Artık gerek duyulmayan Babel bağımlılığı kaldırıldı.
- Dil tanımları güncellendi.
- LDAP ve Celery eklentilerinin belgeleri iyileştirildi.
- Yeni dos-eol ve auto-java-messageformat imlerinin etkinleştirilmesinde yaşanan sorun düzeltildi.
- PyPI paketindeki setup.py sınamasının çalıştırılmasında yaşanan sorun düzeltildi.
- Çoğul biçimlerin işlenmesi iyileştirildi.
- Bazı uç durumlarda yaşanan çeviri yükleme API sorunu düzeltildi.
- El ile değiştirilirken Git yapılandırmasının güncellenmesinde yaşanan sorun düzeltildi.

4.48.21 Weblate 3.2.1

10 Ekim 2018 tarihinde yayınlandı.

- Python 2.7 sürümünde backports.csv belge bağımlılığı eklendi.
- Sınamaların Root olarak çalıştırılması sorunu düzeltildi.
- Gitexport modülünde hataların işlenmesi iyileştirildi.
- Yeni eklenen dillerin ilerleme raporlaması düzeltildi.
- Celery işlemi hatalarının Sentry üzerine doğru şekilde bildirilmesi sağlandı.
- Qt Linguist ile yeni çeviriler oluştururken yaşanan sorun düzeltildi.
- Ara sıra yaşanan tam metin dizini güncelleme sorunları düzeltildi.
- Yeni bileşenler oluşturulurken doğrulama iyileştirildi.
- Eski önerilerin temizlenmesi desteği eklendi.

4.48.22 Weblate 3.2

6 Ekim 2018 tarihinde yayınlandı.

- Kendiliğinden eklenti kurulumu için install_addon yönetim komutu eklendi.
- Hız sınırı ayarlarının daha ayrıntılı olması sağlandı.
- Excel dosyalarının dışa ve içe aktarılması desteği eklendi.
- Birden çok bileşen bulma eklentisi olması durumunda bileşen temizleme iyileştirildi.
- Microsoft Terminology makine çevirisi arka ucu yeniden yazıldı.
- Weblate üzerindeki bazı işlemlerin yükü Celery üzerine aktarıldı.
- Arama yetenekleri geliştirildi ve kurallı ifade araması eklendi.
- Youdao Zhiyun API makine çevirisi desteği eklendi.
- Baidu API makine çevirisi desteği eklendi.
- Celery kullanan bütünleşik bakım ve temizleme görevleri eklendi.
- Çeviri yükleme başarımı neredeyse %25 oranında iyileştirildi.
- Yükleme sırasında üst bilgilerin birleştirilmesi özelliği kaldırıldı.
- Özel işleme iletileri özelliği kaldırıldı.
- Düzenleme kipinin (zen/tam) yapılandırılabilmesi sağlandı.
- Hataların Sentry üzerine bildirilmesi desteği eklendi.
- Depolara kendiliğinden güncelleme desteği eklendi.
- Kullanıcılar tarafından proje ve bileşen oluşturma özelliği eklendi.
- İç çeviri belleğinin yapılan çevirileri kendiliğinden kaydetmesi sağlandı.
- Kullanıcı ve projeler için var olan çeviri belleklerini içe aktarma özelliği eklendi.
- Ekran görüntüleri ile ilgili dizgelerin daha iyi yönetilmesi sağlandı.
- Java MessageFormat denetimi desteği eklendi.

Çözülen sorunların ayrıntılı listesine bakmak için [GitHub üzerindeki 3.2 kilometre taşı](#) bölümüne bakabilirsiniz.

4.48.23 Weblate 3.1.1

27 Temmuz 2018 tarihinde yayınlandı.

- Bazı kurulumlarda yaşanan deneme paketi sorunu düzeltildi.

4.48.24 Weblate 3.1

27 Temmuz 2018 tarihinde yayınlandı.

- 3.0.1 öncesi sürümlerden yükseltmeler desteklenmiyor.
- Ayarlar bölümüne varsayılan işleme iletilerini değiştirme özelliği eklendi.
- Web kancalarının kendi kendine barındırma ortamları ile uyumluluğunu iyileştirildi.
- Amazon Translate desteği eklendi.
- Django 2.1 uyumluluğu sağlandı.
- Django sistem denetimlerine kurulumla ilgili sorunları tanılama özelliği eklendi.
- Yakında kapatılacak libavatar hizmetinin desteği kaldırıldı.
- Değiştirilmemiş çevirileri “Düzenlenmesi gerekli” olarak işaretleyen eklenti eklendi.
- Çeviri yaparken belirli bir konuma atlama desteği eklendi.
- Çeviri indirme işleminin özelleştirilebilmesi sağlandı.
- Çeviri belleği eşleşmelerinde dizge benzerliğinin hesaplanması iyileştirildi.
- Git işlemlerini GnuPG tarafından imzalama özelliği eklendi.

4.48.25 Weblate 3.0.1

10 Haziran 2018 tarihinde yayınlandı.

- 2.20 sürümünden geçişte yaşanabilecek sorun düzeltildi.
- Yerelleştirme güncelleştirmeleri yapıldı.
- Kullanımdan kaldırılmış kanca örnekleri silindi.
- Ön bellek belgeleri iyileştirildi.
- Yönetici belgelerinin görüntülenmesinde yaşanan sorun düzeltildi.
- Uzun dil adlarının işlenmesi iyileştirildi.

4.48.26 Weblate 3.0

1 Haziran 2018 tarihinde yayınlandı.

- Erişim denetimi yeniden yazıldı.
- Modüllerin taşınmasına ve yeniden adlandırılmasına yol açan çeşitli kod temizlemeleri yapıldı.
- Yeni kendiliğinden bileşen bulma eklentisi eklendi.
- import_project yönetim komutunun parametreleri zenginleştirildi.
- Windows RC dosyaları için temel destek eklendi.
- Katkıda bulunan adlarını PO dosya üst bilgilerinde kaydeden yeni eklenti eklendi.
- Her bileşene özgü kanca betikleri özelliği kaldırıldı. Bunun yerine eklentiler kullanıldı.
- Katılımcı anlaşmalarını toplama desteği eklendi.

- Erişim denetimi değişikliklerinin geçmiş kayıtlarında görülmesi sağlandı.
- Bir projedeki tüm bileşenlerin çevirilerinin aynı olmasını sağlayan eklenti eklendi.
- İşleme iletisi kalıplarının daha fazla değişkeni desteklemesi sağlandı.
- Ek metin bağlamı yazma desteği eklendi.

4.49 Weblate 2.x serisi

4.49.1 Weblate 2.20

4 Nisan 2018 tarihinde yayınlandı.

- Subversion depolarını kopyalama hızı iyileştirildi.
- Depo kilitleme özelliği, üçüncü taraf kitaplığı kullanacak biçimde değiştirildi.
- Yalnızca işlem gerektiren dizgeleri indirme desteği eklendi.
- Aynı anda birkaç dilde arama yapma özelliği eklendi.
- Gettext çıktı kaydırmasını yapılandıran yeni eklenti eklendi.
- JSON biçimlendirmesini yapılandıran yeni eklenti eklendi.
- RFC 6750 uyumlu Bearer kimlik doğrulaması kullanılarak, API üzerine kimlik doğrulama desteği eklendi.
- Makine çevirisi hizmetlerini kullanan kendiliğinden çeviri özelliği eklendi.
- Beyaz tahta iletilerine HTML kod imi desteği eklendi.
- Dizge durumlarını toplu değiştirme özelliği eklendi.
- En düşük translate-toolkit sürümü 2.3.0 olarak güncellendi. Eski sürümler artık desteklenmiyor.
- İç çeviri belleği eklendi.
- Panoya bileşen listeleri özeti ve her bileşen listesinin özeti sayfaları eklendi.
- DeepL makine çevirisi hizmeti desteği eklendi.
- Makine çevirisi sonuçlarının Weblate ön belleğine alınması sağlandı.
- İşlenmiş değişiklikleri yeniden sıralama özelliği eklendi.

4.49.2 Weblate 2.19.1

20 Şubat 2018 tarihinde yayınlandı.

- 2.18 sürümünden yükseltme sırasında yaşanan aktarım sorunu düzeltildi.
- Dosya yükleme API doğrulaması iyileştirildi.

4.49.3 Weblate 2.19

15 Şubat 2018 tarihinde yayınlandı.

- Bazı dosya biçimlerinde yaşanan içe aktarma sorunları düzeltildi.
- İnsan dostu tarayıcı bilgilerinin denetim günlüğünde görüntülenmesi sağlandı.
- Dosyalar için TMX dışı aktarıcısı eklendi.
- Çeviri dosyalarının yüklenmesinde çeşitli başarımlar iyileştirmeleri yapıldı.
- Django kullanımını öne alacak biçimde Weblate erişim yönetimini devre dışı bırakma seçeneği eklendi.

- Büyük dizgeler için sözlük arama hızı iyileştirildi.
- django_auth_ldap 1.3.0 uyumluluğu sağlandı.
- Yapılandırma hatalarının kaydedilmesi ve kalıcı olarak bildirilmesi sağlandı.
- Beyaz boşluk düzelticisinde imlerin yok sayılmasına uyuldu.
- Bazı Subversion kurulumlarıyla uyumluluk geliştirildi.
- İç makine çevirisi hizmeti iyileştirildi.
- SAP Translation Hub hizmeti desteği eklendi.
- Microsoft Terminology hizmeti desteği eklendi.
- Bildirim e-postalarındaki reklam özelliği kaldırıldı.
- Dil düzeyinde çeviri ilerleme raporları iyileştirildi.
- Farklı çoğul biçim formülleri desteği iyileştirildi.
- stdlayout kullanmayan Subversion depoları desteği eklendi.
- Çeviri iş akışlarını özelleştirme eklentileri eklendi.

4.49.4 Weblate 2.18

15 Aralık 2017 tarihinde yayınlandı.

- Katılımcı istatistikleri zenginleştirildi.
- Özel karakterler sanal klavye yapılandırması iyileştirildi.
- DTD dosya biçimi desteği eklendi.
- Klavye kısayolları, tarayıcı ve sistem kısayolları ile çakışma olasılığını düşürmek için değiştirildi.
- XLIFF dosyaları için onaylandı imi desteği iyileştirildi.
- gettext PO dosyalarında uzun dizgelerin kaydırılmaması özelliği eklendi.
- Geçerli çevirinin kalıcı bağlantısını kopyalama düğmesi eklendi.
- Django 1.10 desteği kaldırıldı ve Django 2.0 desteği eklendi.
- Çeviri sırasında çevirilerin kilitlenmesi özelliği kaldırıldı.
- Tek dilli çevirilere yeni dizgeler ekleme özelliği eklendi.
- Seçilmiş onaylayıcılar ile çeviri iş akışı özelliği eklendi.

4.49.5 Weblate 2.17.1

13 Ekim 2017 tarihinde yayınlandı.

- Bazı özel durumlarda deneme paketinin çalıştırılmasında yaşanan sorun düzeltildi.
- Yereller güncellendi.

4.49.6 Weblate 2.17

13 Ekim 2017 tarihinde yayınlandı.

- Git kopyalarının varsayılan olarak sığ olması sağlandı.
- Büyük çeviri dosyalarını güncelleme başarımı artırıldı.
- Belirli e-posta adresleri ile hesap açılması engelleme özelliği eklendi.
- Kullanıcıların kendi yorumlarını silebilmesi özelliği eklendi.
- Arama ve değiştirme özelliği için ön izleme adımı eklendi.
- Arama ve yükleme formlarındaki ayarların istemci tarafında tutulması sağlandı.
- Arama yetenekleri zenginleştirildi.
- Her projeye özel erişim denetimi yapılandırmasının ayrıntıları artırıldı.
- BASE_DIR varsayılan değeri değiştirildi.
- Yanlışlıkla silmeyi önlemek için hesap silme işleminin iki adımda yapılması sağlandı.
- Proje erişim denetimi ayarlarını düzenleme özelliği eklendi.
- Öneriler için Akismet ile istenmeyen içerikleri engelleme özelliği eklendi.

4.49.7 Weblate 2.16

11 Ağustos 2017 tarihinde yayınlandı.

- Çeşitli başarımla iyileştirmeleri yapıldı.
- İç içe geçmiş JSON biçimi desteği eklendi.
- WebExtension JSON biçimi desteği eklendi.
- Git dışı aktarıcıda yaşanan kimlik doğrulaması sorunu düzeltildi.
- Belirli durumlar için CSV içe aktarma işlemi iyileştirildi.
- Diğer çeviriler pano bileşeni görünümü iyileştirildi.
- En fazla uzunluk denetimlerinin formda metnin uzunluğunu dayatması sağlandı.
- commit_pending yaşının her bileşene özel yapılandırılabilmesi sağlandı.
- Kullanıcı arayüzünde bazı sadeleştirmeler yapıldı.
- Çevirilerin bileşen/proje/site genelinde aramasında yaşanan sorun düzeltildi.

4.49.8 Weblate 2.15

30 Haziran 2017 tarihinde yayınlandı.

- Diğer çevirilerden daha fazla ilgili çevirinin görüntülenmesi sağlandı.
- Geçerli dizgenin diğer dillerdeki çevirilerini görüntüleme özelliği eklendi.
- Litvanyaca için varsayılan olarak 4 çoğul biçimin kullanılması sağlandı.
- Farklı biçimdeki tek dilli dosyalarda yaşanan yükleme sorunu düzeltildi.
- Başarısız olan kimlik doğrulaması hata iletileri iyileştirildi.
- Sözcüğü sözlükten silerken sayfa durumunun korunması sağlandı.
- İkincil dil çevirisini düzenlemek için doğrudan bağlantı eklendi.
- Perl biçimi kalite denetimi eklendi.

- Yeniden kullanılan parolaları reddetme özelliği eklendi.
- Araç çubuğu sağdan sola yazılan dilleri düzenlemek için zenginleştirildi.

4.49.9 Weblate 2.14.1

24 Mayıs 2017 tarihinde yayınlandı.

- Arama sonuçları sayfalandırılırken yaşanabilecek sorun düzeltildi.
- Bazı uç durumlarda yaşanan eski sürümlerden aktarım sorunu düzeltildi.
- Proje izleme ve izlemeyi bırakma işlemlerinde yaşanabilecek CSRF sorunu düzeltildi.
- Parola sıfırlama işleminde kullanıcı kimliğinin doğrulanması özelliği kaldırıldı.
- Parola unutulduğunda yaşanabilecek CAPTCHA atlama sorunu düzeltildi.

4.49.10 Weblate 2.14

17 Mayıs 2017 tarihinde yayınlandı.

- Sözlük kayıtlarının eklenmesinde AJAX kullanıldı.
- Oturumu kapatma işleminde CSRF engellemesi için POST kullanımına geçildi.
- API anahtarı belirtecini sıfırlama işleminde CSRF engellemesi için POST kullanımına geçildi.
- Weblate tarafından, Content-Security-Policy ayarının varsayılan olarak yapılması sağlandı.
- Yerel düzenleyici adresinin, kendi kendine XSS tehlikesini önlemek için doğrulanması sağlandı.
- Parolanın varsayılan olarak bilinen sorunlara karşı doğrulanması sağlandı.
- Kullanıcıların, hesaplarında yapılan parola değişikliği gibi önemli işlemlerle ilgili bilgilendirilmesi sağlandı.
- CSV dış aktarımlarında olabilecek formüllerden kaçınılması sağlandı.
- Güvenlikte bazı küçük iyileştirmeler yapıldı.
- Kimlik doğrulama girişimlerine hız sınırlama özelliği eklendi.
- Geçmişe öneri içeriklerinin kaydedilmesi sağlandı.
- Önemli hesap işlemlerinin denetim günlüğüne kaydedilmesi sağlandı.
- Hesap silme ya da ilişkilendirme işlemlerinde parola onayı istenmesi özelliği eklendi.
- Önerinin yapıldığı zamanın görüntülenmesi sağlandı.
- Sondaki noktalı virgül karakteri kalite denetimi eklendi.
- Arama bağlantılarının paylaşılabildiğinden emin olunması sağlandı.
- API üzerine kaynak dizge bilgileri ve ekran görüntüleri eklendi.
- API yüklemesi ile çevirilerin üzerine yazma özelliği eklendi.

4.49.11 Weblate 2.13.1

12 Nisan 2017 tarihinde yayınlandı.

- Yönetilen projelerin kullanıcı profilinde listelenmesinde yaşanan sorun düzeltildi.
- Bazı izinlerin eksik olmasına neden olan aktarım sorunu düzeltildi.
- Çevirileri indirirken geçerli dosya biçiminin listelenmesinde yaşanan sorun düzeltildi.
- Kullanıcı izinlerinin yetersiz olduğu bir projeye erişmeye çalışıldığında HTTP 404 sayfasının görüntülenmesi sağlandı.

4.49.12 Weblate 2.13

12 Nisan 2017 tarihinde yayınlandı.

- Çeviri kalıplarında yaşanan kalite denetimleri sorunu düzeltildi.
- Çeviri kaybında tetiklenen kalite denetimi eklendi.
- Bekleyen kullanıcı önerilerini görüntüleme seçeneği eklendi.
- Bileşen listelerini kendiliğinden oluşturma seçeneği eklendi.
- Kimliğini doğrulamamış kullanıcılara görüntülenecek varsayılan panonun yapılandırılabilmesi sağlandı.
- Onaylama işlemi için 25 rastgele dizgeye göz atma seçeneği eklendi.
- Geçmişe dizge değişikliği işlemlerinin kaydedilmesi sağlandı.
- Yeni çeviri ekleme sırasındaki hata bildirimleri iyileştirildi.
- Proje içindeki her dile özel arama özelliği eklendi.
- Grup erişim denetimi listeleri belirli izinlerle sınırlandırıldı.
- Her projeye özel ayrı erişim denetimi listesi için grup erişim denetimi listesinin kullanılması sağlandı.
- İzin denetiminin daha ayrıntılı yapılması sağlandı.
- Kullanıcı arayüzünde bazı küçük iyileştirmeler yapıldı.

4.49.13 Weblate 2.12

3 Mart 2017 tarihinde yayınlandı.

- Grup yönetimi arayüzü iyileştirildi.
- Yandex Translate API desteği eklendi.
- Site genelinde arama hızı iyileştirildi.
- Proje ve bileşen kapsamında arama özelliği eklendi.
- Proje ve bileşen kapsamında arama ve değiştirme özelliği eklendi.
- Tutsuz çevirilerin görüntülenmesi iyileştirildi.
- Kaynak dosyaları yerel düzenleyicide açma desteği eklendi.
- Görsel klavyeyi özel karakterlerle yapılandırma desteği eklendi.
- Kaynak dizgeleri eşleştirmek için görsel karakter tanıma (OCR) desteği ile ekran görüntüsü yönetimi iyileştirildi.
- Varsayılan işleme iletisine çeviri bilgileri ve adresi eklendi.
- Joomla! çeviri biçimi desteği eklendi.
- Farklı dosya biçimleri için içe aktarma güvenilirliği iyileştirildi.

4.49.14 Weblate 2.11

31 Ocak 2017 tarihinde yayınlandı.

- Dil sayfasına ayrıntılı dil bilgileri eklendi.
- Mercurial arka ucunda iyileştirmeler yapıldı.
- Çeviri bileşeni önceliğini belirtme özelliği eklendi.
- Daha az kullanılan izinlerle bile Grup erişim denetiminin daha tutarlı kullanılması sağlandı.
- Kanca betiklerine WL_BRANCH değişkeni eklendi.
- Geliştirici belgeleri iyileştirildi.
- Git dışı aktarma eklentisinde çeşitli Git sürümleriyle daha iyi uyumluluk sağlandı.
- Her proje ve her bileşen için ayrı istatistikler özelliği eklendi.
- Microsoft Translate API desteğine iyileştirmek için dil kodu eşleme özelliği eklendi.
- Çeviri silme işleminin daha hızlı yapılabilmesi için tam metin temizleme işlemi arka plan görevine taşındı.
- Tek çoğul biçimi olan dillerde çoğul kaynağın görüntülenme sorunu düzeltildi.
- import_project komutunda hata işleme iyileştirildi.
- Çeşitli başarımlar iyileştirmeleri yapıldı.

4.49.15 Weblate 2.10.1

20 Ocak 2017 tarihinde yayınlandı.

- Parola sıfırlama formundan hesabın varlığı bilgisinin sızdırılması engellendi (CVE-2017-5537).

4.49.16 Weblate 2.10

15 Aralık 2016 tarihinde yayınlandı.

- Çoğul biçimlerin aynı olup olmadığına bakan kalite denetimi eklendi.
- Kimlik doğrulaması kullanan depolarda yaşanan GitHub kancaları sorunu düzeltildi.
- İsteğe bağlı Git dışı aktarma modülü eklendi.
- Microsoft Cognitive Services Translator API desteği eklendi.
- Proje ve bileşen kullanıcı arayüzü basitleştirildi.
- Kontrol karakterlerini kaldıran kendiliğinden düzeltme özelliği eklendi.
- Projelere her dilin özeti görünümü eklendi.
- CSV dışı aktarma özelliği eklendi.
- İstatistiklerin CSV olarak indirilmesi özelliği eklendi.
- Tüm çevirilere hızlı özet için matris görünümü eklendi.
- Değişiklikler ve dizgeler için temel API eklendi.
- Makine çevirileri için Apertium APy sunucusu desteği eklendi.

4.49.17 Weblate 2.9

4 Kasım 2016 tarihinde yayınlandı.

- createadmin yönetim komutu parametreleri zenginleştirildi.
- import_json komutu var olan bileşenleri işlemek için zenginleştirildi.
- YAML dosyaları desteği eklendi.
- Proje sahiplerinin çeviri bileşenini ve proje ayrıntılarını yapılandırması özelliği eklendi.
- Projeler için “Abone olma” terimi “İzleme” terimi ile değiştirildi.
- Projelerin doğrudan proje sayfasından izlenebilmesi sağlandı.
- Çok dilli durum pano bileşeni eklendi.
- Kaynak gösterilmiyorsa ikincil dilin vurgulanması sağlandı.
- Geçmişe öneri silme işlemlerinin kaydedilmesi sağlandı.
- Kullanıcı profilinde dil seçimi arayüzü iyileştirildi.
- Bileşen için beyaz tahta iletilerinin görüntülenmesinde yaşanan sorun düzeltildi.
- Kaydettikten sonra ayarlar sekmesinin seçili tutulması sağlandı.
- Kaynak dizge yorumunun daha belirgin görüntülenmesi sağlandı.
- Git depoları için Gettext PO birleştirme sürücüsünün kendiliğinden kurulması sağlandı.
- Arama ve değiştirme özelliği eklendi.
- Çeviriler için görsel bağlam (ekran görüntüleri) yükleme özelliği eklendi.

4.49.18 Weblate 2.8

31 Ağustos 2016 tarihinde yayınlandı.

- Belgelendirmede iyileştirmeler yapıldı.
- Çeviriler.
- Paketlenmiş JavaScript kitaplıkları güncellendi.
- list_translators yönetim komutu eklendi.
- Django 1.8 desteği eklendi.
- Django 1.10 uyumluluğunda yaşanan sorun düzeltildi.
- Subversion desteği eklendi.
- XML geçerlilik denetimi uyumsuz XML kod imlerinden ayrıldı.
- API, HIDE_REPO_CREDENTIALS ayarlarına uyacak biçimde düzeltildi.
- Zen kipinde kaynak değişikliğinin görüntülenmesi sağlandı.
- Alt + PageUp/PageDown/Home/End kısayol tuşlarının Zen kipinde de çalışması sağlandı.
- Değişikliklerin tam zamanını görüntüleyen ipucu eklendi.
- Çeviri sayfasına süzgeç seçme ve arama yapma seçeneği eklendi.
- Çeviri silme kullanıcı arayüzü eklendi.
- Yerleştirilebilir öğeleri ekleme davranışı iyileştirildi.
- Zen kipinde yaşanan kendiliğinden kilitleme sorunları düzeltildi.

4.49.19 Weblate 2.7

10 Temmuz 2016 tarihinde yayınlandı.

- Google web çeviri makine çeviri hizmeti kaldırıldı.
- Çeviri eklenirken işleme iletisi iyileştirildi.
- İbranice için Google Çeviri API düzeltildi.
- Mercurial 3.8 ile uyumluluk sağlandı.
- import_json yönetim komutu eklendi.
- Listelenen çevirilerin doğru sıralanması sağlandı.
- Yalnızca bir farkın değil, tam öneri metninin görüntülenmesi sağlandı.
- API zenginleştirildi (ayrıntılı depo durumu, istatistikler, ...).
- Deneme paketindeki deneme depoları için ağ erişimi gereksinimi kaldırıldı.

4.49.20 Weblate 2.6

28 Nisan 2016 tarihinde yayınlandı.

- Dil süzgeci ile bileşenlerin doğrulanmasında yaşanan sorun düzeltildi.
- XLIFF dosyaları desteği iyileştirildi.
- İngilizce olmayan kaynakların makine çevirisi sorunu düzeltildi.
- REST API eklendi.
- Django 1.10 uyumluluğu sağlandı.
- Beyaz tahta iletilerine kategori özelliği eklendi.

4.49.21 Weblate 2.5

10 Mart 2016 tarihinde yayınlandı.

- Proje sahipleri için kendiliğinden çeviri düzeltildi.
- İşleme ve itme işlemlerinin başarımı iyileştirildi.
- Komut satırından öneriler eklemek için yeni yönetim komutu eklendi.
- Dosya yüklerken yorumları birleştirme özelliği eklendi.
- C printf biçimindeki bazı GNU eklentileri desteği eklendi.
- Belgelendirmede iyileştirmeler yapıldı.
- Çevirmen kredileri oluşturma özelliği eklendi.
- Katılımcı istatistikleri oluşturma özelliği eklendi.
- Site genelinde aramanın yalnızca bir dilde arama yapması sağlandı.
- Ermenice için kalite denetimleri iyileştirildi.
- Çeviriler var olmadan çeviri bileşenlerini başlatma özelliği eklendi.
- Qt TS ile yeni çeviriler ekleme özelliği eklendi.
- PHP dosyalarını çevirme desteği iyileştirildi.
- Kalite denetimlerinin başarımı iyileştirildi.
- Denetim hatalarının site genelinde aranması sorunu düzeltildi.

- Kaynak dili belirtme seçeneği eklendi.
- XLIFF dosyaları desteği iyileştirildi.
- import_project için seçenekler listesi zenginleştirildi.
- Beyaz tahta iletileri için hedefleme iyileştirildi.
- Projeler arasında kendiliğinden çeviri özelliği eklendi.
- Tam metin arama dizini iyileştirildi.
- Kendiliğinden çeviri için yönetim komutu eklendi.
- Yerleştirilebilir ögeleri vurgulama özelliği eklendi.
- Yerleştirilebilir ögeler, denetimler ve makine çevirileri için klavye kısayolları eklendi.
- Çeviri kilitleme özelliği iyileştirildi.
- AngularJS ara değerlemesi için kalite denetimi eklendi.
- Kapsamlı grup tabanlı erişim denetimi listeleri eklendi.
- Düzenlenmesi gereken dizgeler (eski adıyla belirsiz) terimi açıklığa kavuşturuldu.
- İşlem yapılması gereken ve çevrilmemiş dizgelerle ilgili terimler açıklığa kavuşturuldu.
- Python 3 desteği eklendi.
- Django 1.7 desteği kaldırıldı.
- Yeni gettext PO dosyaları oluşturmak için msginit bağımlılığı kaldırıldı.
- Yapılandırılabilir pano görünümleri eklendi.
- İşleme hatalarıyla ilgili bildirimler iyileştirildi.
- Aynı adlı bileşenleri import_project komutu ile içe aktarma seçeneği eklendi.
- PHP dosyalarını çevirme desteği iyileştirildi.
- Sözlüğü XLIFF olarak dışa aktarma özelliği eklendi.
- Tüm çevirileri XLIFF ve gettext PO olarak dışa aktarma özelliği eklendi.
- Belgelendirmede iyileştirmeler yapıldı.
- Kendiliğinden grup atamalarını yapılandırma özelliği eklendi.
- Yeni çevirilerin eklenmesi iyileştirildi.

4.49.22 Weblate 2.4

20 Eylül 2015 tarihinde yayınlandı.

- PHP dosyaları desteği iyileştirildi.
- Anonim kullanıcıya erişim denetimi listesi ekleme özelliği eklendi.
- import_project komutunun yapılandırılabilirliği iyileştirildi.
- Geçmişin CSV dökümünü alma özelliği eklendi.
- Kopyalama/yapıştırma işlemlerinde boşluk karakteri hataları engellendi.
- Bitbucket web kancaları desteği eklendi.
- Çeviri yüklerken belirsiz dizgeler üzerine uygulanan denetim sıkılaştırıldı.
- Birkaç adres değişti. Yer imlerinizi güncellemeniz gerekebilir.
- Kanca betiklerinin, geçerli klasör olarak sürüm denetimi sistemi kökü ile yürütülmesi sağlandı.
- Kanca betiklerinin, geçerli bileşeni tanımlayan ortam değişkenleriyle yürütülmesi sağlandı.

- Tam metin dizinini en iyi duruma getiren yönetim komutu eklendi.
- Rollbar üzerine hata bildirme desteği eklendi.
- Projelere birden fazla sahip olabilmesi özelliği eklendi.
- Proje sahiplerinin kendilerini yönetebilmesi özelliği eklendi.
- gettext PO ile kullanılan `javascript-format` desteği eklendi.
- XLIFF olarak yeni çeviri ekleme özelliği eklendi.
- Dosya biçiminin kendiliğinden algılanması iyileştirildi.
- Klavye kısayolları zenginleştirildi.
- Birkaç dil için sözlük eşleştirmesi iyileştirildi.
- Sayfaların çoğunun düzeni iyileştirildi.
- Çeviri yaparken sözlüğe sözcük ekleme özelliği eklendi.
- Weblate tarafından yönetilen dilleri süzme özelliği eklendi.
- CSV dosyalarını çevirme ve içe aktarma özelliği eklendi.
- Durağan dosyaların işlenmesi yeniden yazıldı.
- Tek seçenek olduğunda, üçüncü taraf hizmetine doğrudan oturum / hesap açma bağlantılarının sunulması sağlandı.
- Hesap silinirken bekleyen değişikliklerin işlenmesi sağlandı.
- Site adını değiştiren yönetim komutu eklendi.
- Varsayılan işleyeni yapılandırma seçeneği eklendi.
- Yeni çeviri eklenmesinden sonra kancası eklendi.
- Gönderime birden çok dosya eklemek için Ekle seçeneği eklendi.

4.49.23 Weblate 2.3

22 Mayıs 2015 tarihinde yayınlandı.

- Django 1.6 ve South aktarımlarının desteği kaldırıldı.
- Java özellik dosyaları kullanılırken yeni çeviriler ekleme özelliği eklendi.
- Düzenleme yapmadan öneriyi onaylama özelliği eklendi.
- Google OAuth 2.0 desteği iyileştirildi.
- Microsoft .resx dosyaları desteği eklendi.
- Varsayılan robots.txt dosyası çevirilerin kapsamlı olarak taranmasını engelleyecek biçimde ayarlandı.
- Önerileri kabul etme iş akışı basitleştirildi.
- Önemli bildirimleri her zaman alacak proje sahipleri eklendi.
- Tek dilli kalıbın düzenlenmesini devre dışı bırakma özelliği eklendi..
- Depo durumu görünümünde ayrıntılar artırıldı.
- Çeviriyi değiştirirken kalıbı düzenlemek için doğrudan bağlantı eklendi.
- Proje sahiplerine daha fazla izin eklenmesi sağlandı.
- Zen kipinde ikincil dilin görüntülenmesi sağlandı.
- Kaynak dizgenin ikincil dil için gizlenmesi özelliği eklendi.

4.49.24 Weblate 2.2

19 Şubat 2015 tarihinde yayınlandı.

- Başarım iyileştirmeleri yapıldı.
- Konum ve yorum alanlarında tam metin arama özelliği eklendi.
- Yeni SVG/JavaScript temelli etkinlik çizelgeleri eklendi.
- Django 1.8 desteği eklendi.
- Yorumları silme özelliği eklendi.
- Sahip SVG simgesi eklendi.
- Google Analytics desteği eklendi.
- Çeviri dosyası adlarının işlenmesi iyileştirildi.
- Tek dilli JSON çevirileri desteği eklendi.
- Geçmişe bileşen kilitleme işlemlerinin kaydedilmesi sağlandı.
- Tek dilli çeviriler için kaynak (kalıp) dilini düzenleme özelliği eklendi.
- Temel Gerrit desteği eklendi.

4.49.25 Weblate 2.1

5 Aralık 2014 tarihinde yayınlandı.

- Mercurial depoları desteği eklendi.
- Glyphicon yazı tipi Awesome ile değiştirildi.
- Sosyal ağ kimlik doğrulama hizmetlerinin simgeleri eklendi.
- Düğme renklerinin ve simgelerinin tutarlılığı artırıldı.
- Belgelendirmede iyileştirmeler yapıldı.
- Çeşitli hata düzeltmeleri yapıldı.
- Küçük ekranlar için çeviri listesindeki sütunların kendiliğinden gizlenmesi sağlandı.
- Dosya sistemi yollarının yapılandırması değiştirildi.
- SSH anahtarlarının işlenmesi ve depolanması iyileştirildi.
- Depo kilitleme özelliği iyileştirildi.
- Her kaynak dizge için özelleştirilebilir kalite denetimleri eklendi.
- Tamamlanan çevirilerin panoda gizlenmesi sağlanması.

4.49.26 Weblate 2.0

6 Kasım 2014 tarihinde yayınlandı.

- Bootstrap kullanan yeni uyumlu kullanıcı arayüzü eklendi.
- Sürüm denetimi sistemi arka ucu yeniden yazıldı.
- Belgelendirmede iyileştirmeler yapıldı.
- Site genelindeki iletiler için beyaz tahta eklendi.
- Dizgelerin önceğinin yapılandırılabilmesi sağlandı.
- JSON dosya biçimi desteği eklendi.

- Bazı durumlarda mo dosyaları oluşturulurken yaşanan sorun düzeltildi.
- GitLab bildirimleri özelliği eklendi.
- Çeviri önerilerini devre dışı bırakma özelliği eklendi.
- Django 1.7 desteği eklendi.
- Erişim denetimi listesi projelerine kullanıcı yönetimi eklendi.
- Arama olanakları genişletildi.
- Çevirmenlere çoğul biçimler hakkında daha fazla ipucu verilmesi sağlandı.
- Git deposunun kilitlenmesi sorunu düzeltildi.
- Eski Git sürümleriyle uyumluluk sağlandı.
- Erişim denetimi listesi desteği geliştirildi.
- Her dil için tırnak ve diğer özel karakter düğmeleri eklendi.
- İstatistikleri JSONP olarak dışa aktarma özelliği eklendi.

4.50 Weblate 1.x serisi

4.50.1 Weblate 1.9

6 Mayıs 2014 tarihinde yayınlandı.

- Django 1.6 uyumluluğu sağlandı.
- Artık Django 1.4 uyumluluğu korunmuyor.
- Çeviriler için kilitleme ve kilit açma yönetim komutları eklendi.
- Qt TS dosyaları desteği iyileştirildi.
- Kullanıcıların hesaplarını silebilmesi sağlandı.
- Avatarları devre dışı bırakma özelliği eklendi.
- Ad ve soyad öznitelikleri birleştirildi.
- Avatarların sunucu tarafında tutulması ve ön belleğe alınması sağlandı.
- shields.io simgeleri desteği eklendi.

4.50.2 Weblate 1.8

7 Kasım 2013 tarihinde yayınlandı.

- Lütfen yükseltme yönergeleri için rehbera bakın.
- Proje özetinin görünümü güzelleştirildi.
- Paylaşım seçeneklerinin daha iyi görünmesi sağlandı.
- Anonim kullanıcı izinleri üzerinde daha fazla denetim sağlandı.
- Üçüncü taraf hizmetlerini kullanarak oturum açma desteği eklendi. Bilgi almak için rehbera bakabilirsiniz.
- Kullanıcı adı yerine e-posta ile oturum açılabilmesi sağlandı.
- Belgelendirmede iyileştirmeler yapıldı.
- Kaynak dizge onaylama işlemi geliştirildi.
- Tüm dizgelerde arama özelliği eklendi.

- Kaynak dizgelerin izlenmesi iyileştirildi.
- Hesap açılışına Captcha koruması eklendi.

4.50.3 Weblate 1.7

7 Ekim 2013 tarihinde yayınlandı.

- Lütfen yükseltme yönergeleri için rehbera bakın.
- Python parantezi biçim dizgesini denetleme desteği eklendi.
- Kalite denetimlerinin her bileşen için ayrı özelleştirilmesi sağlandı.
- Çeviri istatistiklerine göre ayrıntılar eklendi.
- Öneri, denetim ve yorumların dizgelerle ilişkilendirme yöntemi değiştirildi.
- Kullanıcıların gönderim iletisine metin ekleyebilmesi sağlandı.
- Yeni dil isteklerine abone olma özelliği eklendi.
- Yeni çeviriler ekleme özelliği eklendi.
- Pano bileşenleri ve çizelgelerde Pango + Cairo yerine Pillow kullanılması sağlandı.
- Durum simgesi pano bileşeni eklendi.
- Geçersiz metin yönü denetimi kaldırıldı.
- Sözlükte yapılan değişiklikler geçmiş kayıtlarına eklendi.
- Çeviri görünümü başarıımı iyileştirildi.

4.50.4 Weblate 1.6

25 Temmuz 2013 tarihinde yayınlandı.

- Hesap açarken çıkabilecek sorunların işlenmesi iyileştirildi.
- Değişikliklere göz atma özelliği eklendi.
- Makine çevirisi önerilerinin sıralanması düzeltildi.
- MyMemory makine çevirisi desteği iyileştirildi.
- Amagama makine çevirisi desteği eklendi.
- Sık kullanılan sayfalarda çeşitli iyileştirmeler yapıldı.
- Arama sonuçlarında aranan ifadenin vurgulanması sağlandı.
- İleti kaydedilirken kendiliğinden düzeltme özelliği eklendi.
- Çeviri geçmişini izleme ve geri döndürme özellikleri eklendi.
- Google Çeviri API desteği eklendi.
- SSH sunucu anahtarları yönetimi desteği eklendi.
- Çeşitli form doğrulama iyileştirmeleri yapıldı.
- Çeşitli kalite denetimi iyileştirmeleri yapıldı.
- İçerik aktarma başarıımı iyileştirildi.
- Önerilere oy verme desteği eklendi.
- Yönetim arayüzü sadeleştirildi.

4.50.5 Weblate 1.5

16 Nisan 2013 tarihinde yayınlandı.

- Lütfen yükseltme yönergeleri için rehbera bakın.
- Herkese açık kullanıcı sayfaları eklendi.
- Çoğul biçimlerin adlandırılması iyileştirildi.
- Sözlüğün TBX olarak dışa aktarılması desteği eklendi.
- Bitbucket bildirimleri desteği eklendi.
- Etkinlik çizelgelerinin her çeviri, dil ya da kullanıcı için görülebilmesi sağlandı.
- import_project yönetici komutunun seçenekleri zenginleştirildi.
- Django 1.5 uyumluluğu sağlandı.
- Avatarların libavatar kullanılarak görüntülenmesi sağlandı.
- JSON dışa aktarımının güzel biçimde yazdırılabilmesi sağlandı.
- Çeşitli başarımlar iyileştirmeleri yapıldı.
- Başarısız olan denetimlerin veya belirsiz dizgelerin projelerin veya dillerin ilerleme çubuklarında görüntülenmesi sağlandı.
- Özel ön gönderim kancaları ve ek dosyaların gönderilmesi özellikleri eklendi.
- Arama özelliği daha iyi başarımlar ve kullanıcı deneyimi sağlamak için yeniden yazıldı.
- Makine çevirileri arayüzü yenilendi.
- Tek dilli po dosyaları desteği eklendi.
- Çeşitli aramaların hızını artırmak için ön belleğe alınmış üst veri miktarı artırıldı.
- Sözcük sayılarının görüntülenmesi sağlandı.

4.50.6 Weblate 1.4

23 Ocak 2013 tarihinde yayınlandı.

- Dizge silme işlemiyle ilgili denetimlerin ve yorumların silinmesinde yaşanan sorun düzeltildi.
- Çevirilerin kendiliğinden yayılmasını engelleme seçeneği eklendi.
- Birleştirme hatalarına abone olma seçeneği eklendi.
- Özel ttkit yükleyiciye gerek duyan projelerin doğru biçimde içe aktarılması sağlandı.
- Arama motoru robotlarının erişimini kolaylaştırmak için site haritaları eklendi.
- Bildirim e-postalarına veya özet akışlarına doğrudan dizge bağlantısı eklendi.
- Yönetim arayüzünde çeşitli iyileştirmeler yapıldı.
- Yönetim arayüzünde üretim ortamı kurulumu için ipuçları sunuldu.
- Her dil için ayrı pano bileşeni ve etkileşim sayfası eklendi.
- Çeviri kilitleme işlemi iyileştirildi.
- Pano bileşenleri için daha fazla kod parçası çeşidinin görüntülenmesi sağlandı.
- Başarısız olan denetimlerin veya belirsiz dizgelerin ilerleme çubuklarında belirtilmesi sağlandı.
- Gönderim iletisine daha fazla biçimlendirme seçeneği eklendi.
- Makine çevirisi hizmetleriyle ilgili hataların işlenmesinde yaşanan sorun düzeltildi.
- Kendiliğinden çeviri kilitleme davranışı iyileştirildi.

- Önceki kaynak dizgeye göre yapılan değişikliklerin görüntülenmesi sağlandı.
- Alt dizge arama özelliği eklendi.
- Çeşitli kalite denetimi iyileştirmeleri yapıldı.
- Her proje için erişim denetimi listesi özelliği eklendi.
- Birim sınamalarına göre temel kod kapsamı.

4.50.7 Weblate 1.3

16 Kasım 2012 tarihinde yayınlandı.

- PostgreSQL veri tabanı arka ucuyla uyumluluk sağlandı.
- Yukarı akış Git deposunda silinen dillerin kaldırılması sağlandı.
- Kalite denetimlerinin işlenmesi iyileştirildi.
- Yeni denetimler eklendi (BBCode, XML biçimlendirmesi ve yeni satır karakterleri).
- Birleştirme yerine isteğe bağlı yeniden temellendirme desteği sağlandı.
- Weblate kurulumunun yerinin değiştirilmesi sağlandı (örneğin /weblate yolu altında çalıştırmak için).
- Kendiliğinden algılamının başarısız olması durumunda dosya türünü el ile seçme özelliği eklendi.
- Android kaynakları için daha iyi destek sağlandı.
- Web arayüzünden SSH anahtarı oluşturma özelliği eklendi.
- Veri dışı aktarımlarının daha görünür olması sağlandı.
- Bazı özel karakterleri yazmak için yeni düğmeler eklendi.
- Sözlüğü dışı aktarma özelliği eklendi.
- Tüm Weblate kurulumunu kilitleme özelliği eklendi.
- Kaynak dizge denetimleri ve kaynak dizge onaylama özellikleri eklendi.
- Hem çeviriler hem de kaynak dizgeler için kullanıcı yorumları özelliği eklendi.
- Günlük izleme iyileştirildi.
- Değişikliklerin RSS ile izlenebilmesi sağlandı.
- Sağdan sola yazılan dillerin desteği iyileştirildi.

4.50.8 Weblate 1.2

14 Ağustos 2012 tarihinde yayınlandı.

- Weblate veri tabanı aktarımı için South kullanımına geçildi. Yükseltme yapıyorsanız lütfen yükseltme yönergelerine bakın.
- Bağlı Git depolarıyla ilgili yaşanan küçük sorunlar düzeltildi.
- Giriş sayfası Weblate kullanarak çeviri yapan kişilerin ilgisini çekecek biçimde yenilendi.
- Çeviri projelerini tanıtmak için kullanılabilecek pano bileşenleri eklendi.
- Depoyu özgün durumuna sıfırlama seçeneği eklendi (izin verilmiş kullanıcılar için).
- Proje ve bileşenlere çevirileri kilitleme özelliği eklendi.
- Bazı çevirilerin devre dışı bırakılabilmesi sağlandı.
- Yeni çevirilerin eklenmesine yapılandırılabilir seçenekler eklendi.
- Her proje için Git işlemlerini yapılandırma özelliği eklendi.

- Basit antispam koruması eklendi.
- Ana sayfanın düzeni iyileştirildi.
- Her işlemede değişikliklerin kendiliğinden itilmesi sağlandı.
- Çevirmenlere e-posta bildirimleri gönderme özelliği eklendi.
- Ayarlarda yalnızca kullanılan dillerin listelenmesi sağlandı.
- Proje içe aktarılırken bilinmeyen dillerin işlenmesi iyileştirildi.
- Çevirmen tarafından çeviriyi kitleme özelliği eklendi.
- İsteğe bağlı olarak po dosyasında “Language-Team” üst bilgisinin korunması sağlandı.
- Hakkında sayfasına bazı istatistikler eklendi.
- Django-registration 0.8 sürümü desteği eklendi (ve zorunlu kılındı).
- Denetim hatası olan dizge sayısının ön belleğe alınması sağlandı.
- Kurulum sırasında gereksinimlerin denetlenmesi sağlandı.
- Belgelendirmede iyileştirmeler yapıldı.

4.50.9 Weblate 1.1

4 Temmuz 2012 tarihinde yayınlandı.

- Birkaç çeviri iyileştirildi.
- Bileşen oluştururken doğrulama iyileştirildi.
- Bileşenler arasında paylaşılan Git depoları desteği eklendi.
- Uzak depoyu çekmek için her girişimin işlenmesinin gerekmemesi sağlandı.
- Dizin oluşturma yükünü aktarma desteği eklendi.

4.50.10 Weblate 1.0

10 Mayıs 2012 tarihinde yayınlandı.

- Bileşen eklenirken ve kaydedilirken doğrulama iyileştirildi.
- Android bileşen dosyaları için deneysel destek eklendi (yamanmış tkit gerekir).
- Kancalardan gelen güncellemelerin arka planda yapılması sağlandı.
- Yükleme yönergeleri iyileştirildi.
- Sözlükte gezinme iyileştirildi.

4.51 Weblate 0.x serisi

4.51.1 Weblate 0.9

18 Nisan 2012 tarihinde yayınlandı.

- Bilinmeyen diller içe aktarılırken yaşanan sorun düzeltildi.
- Yakındaki iletilerin listelenmesi iyileştirildi.
- Birkaç denetim iyileştirildi.
- Belgeler güncellendi.

- Birkaç dilin tanımları eklendi.
- Çeşitli kod temizlemeleri yapıldı.
- Belgelendirmede iyileştirmeler yapıldı.
- Dosya düzeni değiştirildi.
- Yardım betikleri Django 1.4 sürümüne güncellendi.
- Çeviri sırasında gezinme iyileştirildi.
- PO dosyası ad değişikliklerinin işlenmesi iyileştirildi.
- Bileşen oluştururken doğrulama iyileştirildi.
- syncdb ile bütünleşik tam kurulum sağlandı.
- Tüm çeviri sayfalarına yapılan son değişikliklerin listesi eklendi.
- Çevrilmemiş dizge denetiminin, yalnızca biçim dizgesi iletilerini yok sayması sağlandı.

4.51.2 Weblate 0.8

3 Nisan 2012 tarihinde yayınlandı.

- Tam metin araması Whoosh ile değiştirildi.
- Denetimlerde çeşitli düzeltme ve iyileştirmeler yapıldı.
- Yeni updatechecks komutu eklendi.
- Çok sayıda çeviri güncellemesi yapıldı.
- Sık kullanılan terimlerin kaydedilebilmesi için sözlük eklendi.
- Depoların durumu özeti için /admin/report/ eklendi.
- Makine çevirisi hizmetlerinin sayfa yüklemesini engellemesi önendi.
- Yönetim arayüzüne yararlı verileri güncelleme işlemleri eklendi.
- Kullanıcılar tarafından yapılan değişikliklerin günlük kayıtlarının tutulması sağlandı.
- Tek bir kullanıcıdan daha az gönderim oluşturmak için Git işlemlerini erteleme özelliği eklendi.
- Başarısız olan denetimlere göz atma özelliği eklendi.
- Önceden çevrilmiş dizgeleri kullanan kendiliğinden çeviri özelliği eklendi.
- Kullanılan sürümleri görüntüleyen, yeni Hakkında sayfası eklendi.
- Django 1.4 uyumluluğu sağlandı.
- Değişiklikleri web arayüzünden uzak depoya itme özelliği eklendi.
- Başkaları tarafından yapılmış çevirileri gözden geçirme özelliği eklendi.

4.51.3 Weblate 0.7

16 Şubat 2012 tarihinde yayınlandı.

- GitHub bildirimlerinin doğrudan desteklenmesi sağlandı.
- Sahipsiz denetim ve çevirileri temizleme desteği eklendi.
- Çeviri sırasında yakındaki dizgeleri görüntüleme özelliği eklendi.
- Çeviri sırasında benzer dizgeleri görüntüleme özelliği eklendi.
- Dizge araması iyileştirildi.

4.51.4 Weblate 0.6

14 Şubat 2012 tarihinde yayınlandı.

- Çevrilmiş iletiler için çeşitli denetimler eklendi.
- Ayarlanabilir erişim denetimi özelliği eklendi.
- Yeni satır karakterleri bulunan çevirilerin işlenmesi iyileştirildi.
- Tabloların istemci tarafında sıralanması özelliği eklendi.
- Yükseltme yapıyorsanız lütfen yükseltme yönergelerini inceleyin.

4.51.5 Weblate 0.5

12 Şubat 2012 tarihinde yayınlandı.

- **Şu çevrimiçi hizmetleri kullanan makine çevirisi desteği eklendi:**
 - Apertium
 - Microsoft Translator
 - MyMemory
- Birkaç yeni çeviri.
- Yukarı akış değişikliklerinin birleştirilmesi iyileştirildi.
- Eşzamanlı Git çekme ve çevirilerinin işlenmesi iyileştirildi.
- Belirsiz değişikliklerin de yayılması sağlandı.
- Yüklenen dosyalardaki çevirilerin de yayılması sağlandı.
- FastCGI (ve büyük olasılıkla diğerleri) kullanılırken dosya indirmelerinde yaşanan sorun düzeltildi.

4.51.6 Weblate 0.4

8 Şubat 2012 tarihinde yayınlandı.

- Belgelere kullanım rehberi eklendi.
- API kancaları CSRF koruması gerektirmeyecek biçimde düzeltildi.

4.51.7 Weblate 0.3

8 Şubat 2012 tarihinde yayınlandı.

- Çoğul çevirilerde kaynağın daha iyi görüntülenmesi sağlandı.
- Sphinxs biçiminde yeni belgeler hazırlandı.
- Çeviri yapılırken ikincil dillerin görüntülenmesi sağlandı.
- Hata sayfası var olan projelerin listesini verecek biçimde geliştirildi.
- Her dil için istatistikler eklendi.

4.51.8 Weblate 0.2

7 Şubat 2012 tarihinde yayınlandı.

- Çeşitli formların doğrulaması iyileştirildi.
- Profili yükseltildiğinde kullanıcıların uyarılması sağlandı.
- Oturum açma adresinin anımsanması sağlandı.
- Çoğul biçimler yazılırken metin aralarının adlandırılması sağlandı.
- Çeviri alanının kendiliğinden genişletilmesi sağlandı.

4.51.9 Weblate 0.1

6 Şubat 2012 tarihinde yayınlandı.

- İlk sürüm.

W

wlc, [159](#)
wlc.config, [160](#)
wlc.main, [160](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component:slug)/, 134
ANY /, 105	GET /api/components/(string:project)/(string:component:slug)/, 130
/api	GET /api/components/(string:project)/(string:component:slug)/, 132
GET /api/, 108	GET /api/components/(string:project)/(string:component:slug)/, 132
/api/addons	GET /api/components/(string:project)/(string:component:slug)/, 131
GET /api/addons/, 146	GET /api/components/(string:project)/(string:component:slug)/, 129
GET /api/addons/(int:id)/, 146	GET /api/components/(string:project)/(string:component:slug)/, 134
PUT /api/addons/(int:id)/, 147	GET /api/components/(string:project)/(string:component:slug)/, 132
DELETE /api/addons/(int:id)/, 147	POST /api/components/(string:project)/(string:component:slug)/, 146
PATCH /api/addons/(int:id)/, 147	POST /api/components/(string:project)/(string:component:slug)/, 134
/api/changes	POST /api/components/(string:project)/(string:component:slug)/, 130
GET /api/changes/, 143	POST /api/components/(string:project)/(string:component:slug)/, 131
GET /api/changes/(int:id)/, 143	POST /api/components/(string:project)/(string:component:slug)/, 132
/api/component-lists	PUT /api/components/(string:project)/(string:component:slug)/, 128
GET /api/component-lists/, 147	DELETE /api/components/(string:project)/(string:component:slug)/, 129
GET /api/component-lists/(str:slug)/, 147	DELETE /api/components/(string:project)/(string:component:slug)/, 134
POST /api/component-lists/(str:slug)/components/, 148	PATCH /api/components/(string:project)/(string:component:slug)/, 127
PUT /api/component-lists/(str:slug)/, 148	/api/groups
DELETE /api/component-lists/(str:slug)/, 148	GET /api/groups/, 112
DELETE /api/component-lists/(str:slug)/components/(string:component:slug)/, 148	GET /api/groups/(int:id)/, 112
PATCH /api/component-lists/(str:slug)/, 148	POST /api/groups/, 112
/api/components	POST /api/groups/(int:id)/changes/, 115
GET /api/components/, 125	POST /api/groups/(int:id)/components/, 114
GET /api/components/(string:project)/(string:component:slug)/, 125	
GET /api/components/(string:project)/(string:component:slug)/, 129	
GET /api/components/(string:project)/(string:component:slug)/, 129	

POST /api/groups/(int:id)/languages/, 115 PUT /api/projects/(string:project)/, 120
 POST /api/groups/(int:id)/projects/, 114 DELETE /api/projects/(string:project)/, 120
 POST /api/groups/(int:id)/roles/, 114 PATCH /api/projects/(string:project)/, 119
 PUT /api/groups/(int:id)/, 113
 DELETE /api/groups/(int:id)/, 114
 DELETE /api/groups/(int:id)/componentlist/, 115
 DELETE /api/groups/(int:id)/components/(int:component_id)/, 114
 DELETE /api/groups/(int:id)/languages/(string:language_code_id)/, 115
 DELETE /api/groups/(int:id)/projects/(int:project_id)/, 114
 PATCH /api/groups/(int:id)/, 113

/api/languages

GET /api/languages/, 116
 GET /api/languages/(string:language)/, 117
 GET /api/languages/(string:language)/statistics/, 118
 POST /api/languages/, 116
 PUT /api/languages/(string:language)/, 117
 DELETE /api/languages/(string:language)/, 118
 PATCH /api/languages/(string:language)/, 117

/api/memory

GET /api/memory/, 141
 DELETE /api/memory/(int:memory_object_id)/, 141

/api/metrics

GET /api/metrics/, 149

/api/projects

GET /api/projects/, 119
 GET /api/projects/(string:project)/, 119
 GET /api/projects/(string:project)/changes/, 120
 GET /api/projects/(string:project)/components/, 121
 GET /api/projects/(string:project)/languages/, 124
 GET /api/projects/(string:project)/repositories/, 120
 GET /api/projects/(string:project)/statistics/, 124
 POST /api/projects/, 119
 POST /api/projects/(string:project)/components/, 121
 POST /api/projects/(string:project)/repositories/, 120

/api/roles

GET /api/roles/, 115
 POST /api/roles/, 115
 DELETE /api/roles/(int:id)/, 116
 PATCH /api/roles/(int:id)/, 116

/api/screenshots

GET /api/screenshots/, 144
 GET /api/screenshots/(int:id)/, 144
 GET /api/screenshots/(int:id)/file/, 144
 POST /api/screenshots/, 145
 POST /api/screenshots/(int:id)/file/, 144
 POST /api/screenshots/(int:id)/units/, 145
 PUT /api/screenshots/(int:id)/, 146
 DELETE /api/screenshots/(int:id)/, 146
 DELETE /api/screenshots/(int:id)/units/(int:unit_id)/, 145
 PATCH /api/screenshots/(int:id)/, 145

/api/tasks

GET /api/tasks/, 149
 GET /api/tasks/(str:uuid)/, 149

/api/translations

GET /api/translations/, 135
 GET /api/translations/(string:project)/(string:component)/, 135
 GET /api/translations/(string:project)/(string:component)/, 137
 GET /api/translations/(string:project)/(string:component)/, 139
 GET /api/translations/(string:project)/(string:component)/, 139
 GET /api/translations/(string:project)/(string:component)/, 140
 GET /api/translations/(string:project)/(string:component)/, 137
 POST /api/translations/(string:project)/(string:component)/, 138
 POST /api/translations/(string:project)/(string:component)/, 139
 POST /api/translations/(string:project)/(string:component)/, 140
 POST /api/translations/(string:project)/(string:component)/, 138

DELETE /api/translations/(string:project)/(string:component)/(string:language)/, 137
POST /hooks/complaint/, 150
POST /hooks/pagure/, 151

/api/units

GET /api/units/, 141
GET /api/units/(int:id)/, 141
PUT /api/units/(int:id)/, 142
DELETE /api/units/(int:id)/, 143
PATCH /api/units/(int:id)/, 142

/api/users

GET /api/users/, 109
GET /api/users/(str:username)/, 109
GET /api/users/(str:username)/notifications/, 111
GET /api/users/(str:username)/notifications/(int:subscription_id)/, 111
GET /api/users/(str:username)/statistics/, 111
POST /api/users/, 109
POST /api/users/(str:username)/groups/, 110
POST /api/users/(str:username)/notifications/, 111
PUT /api/users/(str:username)/, 110
PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 111
DELETE /api/users/(str:username)/, 110
DELETE /api/users/(str:username)/groups/, 110
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 112
PATCH /api/users/(str:username)/, 110
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 112

/exports

GET /exports/rss/, 153
GET /exports/rss/(string:project)/, 153
GET /exports/rss/(string:project)/(string:component)/, 153
GET /exports/rss/(string:project)/(string:component)/(string:language)/, 153
GET /exports/rss/language/(string:language)/, 153
GET /exports/stats/(string:project)/(string:component)/, 152

/hooks

GET /hooks/update/(string:project)/, 150
GET /hooks/update/(string:project)/(string:component)/, 150
POST /hooks/azure/, 151
POST /hooks/bitbucket/, 150
POST /hooks/gitea/, 151
POST /hooks/gitee/, 151
POST /hooks/github/, 150

Semboller

.XML resource file
 file format, [90](#)

--add
 auto_translate komut satırı seçeneği, [385](#)

--addon
 install_addon komut satırı seçeneği, [392](#)

--age
 commit_pending komut satırı seçeneği, [386](#)

--author
 add_suggestions komut satırı seçeneği, [385](#)

--author-email
 wlc komut satırı seçeneği, [157](#)

--author-name
 wlc komut satırı seçeneği, [157](#)

--base-file-template
 import_project komut satırı seçeneği, [390](#)

--check
 importusers komut satırı seçeneği, [391](#)

--config
 wlc komut satırı seçeneği, [155](#)

--config-section
 wlc komut satırı seçeneği, [155](#)

--configuration
 install_addon komut satırı seçeneği, [392](#)

--convert
 wlc komut satırı seçeneği, [156](#)

--email
 createadmin komut satırı seçeneği, [387](#)

--file-format
 import_project komut satırı seçeneği, [390](#)

--force
 loadpo komut satırı seçeneği, [393](#)

--force-commit
 pushgit komut satırı seçeneği, [394](#)

--format
 wlc komut satırı seçeneği, [155](#)

--fuzzy
 wlc komut satırı seçeneği, [157](#)

--ignore
 import_json komut satırı seçeneği, [388](#)

--inconsistent
 auto_translate komut satırı seçeneği, [385](#)

--input
 wlc komut satırı seçeneği, [156](#)

--key
 wlc komut satırı seçeneği, [155](#)

--lang
 loadpo komut satırı seçeneği, [393](#)

--language-code
 list_translators komut satırı seçeneği, [392](#)

--language-map
 import_memory komut satırı seçeneği, [389](#)

--language-regex
 import_project komut satırı seçeneği, [390](#)

--license
 import_project komut satırı seçeneği, [390](#)

--license-url
 import_project komut satırı seçeneği, [390](#)

--main-component
 import_json komut satırı seçeneği, [388](#)

--method
 import_project komut satırı seçeneği, [390](#)

--mode
 auto_translate komut satırı seçeneği, [386](#)

--mt

auto_translate komut satırı seçeneği, 385

--name
createadmin komut satırı seçeneği, 387

--name-template
import_project komut satırı seçeneği, 390

--new-base-template
import_project komut satırı seçeneği, 390

--no-password
createadmin komut satırı seçeneği, 387

--no-privs-update
setupgroups komut satırı seçeneği, 394

--no-projects-update
setupgroups komut satırı seçeneği, 394

--no-update
setuplang komut satırı seçeneği, 395

--output
wlc komut satırı seçeneği, 156

--overwrite
auto_translate komut satırı seçeneği, 385
wlc komut satırı seçeneği, 156

--password
createadmin komut satırı seçeneği, 387

--project
import_json komut satırı seçeneği, 388

--source
auto_translate komut satırı seçeneği, 385

--threshold
auto_translate komut satırı seçeneği, 386

--update
createadmin komut satırı seçeneği, 387
import_json komut satırı seçeneği, 388
install_addon komut satırı seçeneği, 392

--url
wlc komut satırı seçeneği, 155

--user
auto_translate komut satırı seçeneği, 385

--username
createadmin komut satırı seçeneği, 387

--vcs
import_project komut satırı seçeneği, 390

A

add_suggestions
weblate admin command, 385

add_suggestions komut satırı seçeneği
--author, 385

ADMINS
setting, 211

AKISMET_API_KEY
setting, 342

ALLOWED_HOSTS
setting, 211

Android
file format, 84

ANONYMOUS_USER_NAME
setting, 342

API, 105, 153, 158

Apple strings
file format, 85

ARB
file format, 88

AUDITLOG_EXPIRY
setting, 342

AUTH_LOCK_ATTEMPTS
setting, 342

AUTH_TOKEN_VALID
setting, 343

auto_translate
weblate admin command, 385

auto_translate komut satırı seçeneği
--add, 385
--inconsistent, 385
--mode, 386
--mt, 385
--overwrite, 385
--source, 385
--threshold, 386
--user, 385

AUTO_UPDATE
setting, 343

AUTOFIX_LIST
setting, 344

AVATAR_URL_PREFIX
setting, 343

B

BACKGROUND_TASKS
setting, 344

BaseAddon (*weblate.addons.base* içindeki sınıf), 430

BASIC_LANGUAGES
setting, 345

bilingual
translation, 75

BITBUCKETSERVER_CREDENTIALS
setting, 353

BORG_EXTRA_ARGS
setting, 345

C

CACHE_DIR
 setting, 345

can_install() (*weblate.addons.base.BaseAddon sınıfı yöntemi*), 430

CELERY_BACKUP_OPTIONS, 166, 184

CELERY_BEAT_OPTIONS, 166, 184

CELERY_MAIN_OPTIONS, 166, 184

CELERY_MEMORY_OPTIONS, 166, 184

CELERY_NOTIFY_OPTIONS, 166, 184

celery_queues
 weblate admin command, 386

CELERY_TRANSLATE_OPTIONS, 166, 184

changes
 wlc komut satırı seçeneği, 156

CHECK_LIST
 setting, 346

checkgit
 weblate admin command, 386

cleanup
 wlc komut satırı seçeneği, 156

cleanup_ssh_keys
 weblate admin command, 387

cleanuptrans
 weblate admin command, 387

Comma separated values
 file format, 90

Command (*wlc.main içindeki sınıf*), 160

COMMENT_CLEANUP_DAYS
 setting, 346

commit
 wlc komut satırı seçeneği, 155

commit_pending
 weblate admin command, 386

commit_pending komut satırı seçeneği
 --age, 386

COMMIT_PENDING_HOURS
 setting, 347

commitgit
 weblate admin command, 386

configure() (*weblate.addons.base.BaseAddon yöntemi*), 430

CONTACT_FORM
 setting, 347

createadmin
 weblate admin command, 387

createadmin komut satırı seçeneği
 --email, 387
 --name, 387
 --no-password, 387
 --password, 387
 --update, 387
 --username, 387

CSP_CONNECT_SRC
 setting, 346

CSP_FONT_SRC
 setting, 346

CSP_IMG_SRC

setting, 346

CSP_SCRIPT_SRC

setting, 346

CSP_STYLE_SRC

setting, 346

CSV

file format, 90

D

daily() (*weblate.addons.base.BaseAddon yöntemi*), 430

DATA_DIR

setting, 347

DATABASE_BACKUP

setting, 348

DATABASES

setting, 211

DEBUG

setting, 211

DEFAULT_ACCESS_CONTROL

setting, 348

DEFAULT_ADD_MESSAGE

setting, 349

DEFAULT_ADDON_MESSAGE

setting, 349

DEFAULT_ADDONS

setting, 349

DEFAULT_AUTO_WATCH

setting, 348

DEFAULT_COMMIT_MESSAGE

setting, 349

DEFAULT_COMMITTER_EMAIL

setting, 349

DEFAULT_COMMITTER_NAME

setting, 350

DEFAULT_DELETE_MESSAGE

setting, 349

DEFAULT_FROM_EMAIL

setting, 211

DEFAULT_LANGUAGE

setting, 350

DEFAULT_MERGE_MESSAGE

setting, 349

DEFAULT_MERGE_STYLE

setting, 350

DEFAULT_PAGE_LIMIT

setting, 358

DEFAULT_PULL_MESSAGE

setting, 351

DEFAULT_RESTRICTED_COMPONENT

setting, 349

DEFAULT_SHARED_TM

setting, 350

DEFAULT_TRANSLATION_PROPAGATION

setting, 350

download

wlc komut satırı seçeneği, 156

DTD

- file format, 92
- dump_memory
 - weblate admin command, 388
- dumpuserdata
 - weblate admin command, 388

E

- ENABLE_AVATARS
 - setting, 351
- ENABLE_HOOKS
 - setting, 351
- ENABLE_HTTPS
 - setting, 351
- ENABLE_SHARING
 - setting, 351
- EXTRA_HTML_HEAD
 - setting, 352

F

- file format
 - .XML resource file, 90
 - Android, 84
 - Apple strings, 85
 - ARB, 88
 - Comma separated values, 90
 - CSV, 90
 - DTD, 92
 - gettext, 78
 - go-i18n, 88
 - gotext, 88
 - GWT properties, 82
 - i18next, 87
 - INI translations, 82, 83
 - Java properties, 81
 - Joomla translations, 83
 - JSON, 86
 - mi18n lang, 82
 - PHP strings, 85
 - PO, 78
 - Qt, 83
 - RC, 93
 - ResourceDictionary, 90
 - RESX, 90
 - Ruby YAML, 92
 - Ruby YAML Ain't Markup Language, 92
 - string resources, 84
 - TS, 83
 - WPF, 90
 - XLIFF, 79
 - XML, 93
 - YAML, 91
 - YAML Ain't Markup Language, 91

G

- get() (*wlc. Weblate yöntemi*), 159
- get_add_form() (*weblate.addons.base.BaseAddon sınıf yöntemi*), 430
- GET_HELP_URL

- setting, 352
- get_settings_form() (*weblate.addons.base.BaseAddon yöntemi*), 430
- gettext
 - file format, 78
- GITEA_CREDENTIALS
 - setting, 352
- GITHUB_CREDENTIALS
 - setting, 353
- GITLAB_CREDENTIALS
 - setting, 352
- go-i18n
 - file format, 88
- GOOGLE_ANALYTICS_ID
 - setting, 353
- gotext
 - file format, 88
- GWT properties
 - file format, 82

H

- HIDE_REPO_CREDENTIALS
 - setting, 353
- HIDE_VERSION
 - setting, 354

I

- i18next
 - file format, 87
- import_demo
 - weblate admin command, 388
- import_json
 - weblate admin command, 388
- import_json komut satırı seçeneği
 - ignore, 388
 - main-component, 388
 - project, 388
 - update, 388
- import_memory
 - weblate admin command, 389
- import_memory komut satırı seçeneği
 - language-map, 389
- import_project
 - weblate admin command, 389
- import_project komut satırı seçeneği
 - base-file-template, 390
 - file-format, 390
 - language-regex, 390
 - license, 390
 - license-url, 390
 - main-component, 390
 - name-template, 390
 - new-base-template, 390
 - vcs, 390
- importuserdata
 - weblate admin command, 391
- importusers
 - weblate admin command, 391

importusers komut satırı seçeneği
--check, 391

INI translations
file format, 82, 83

install_addon
weblate admin command, 392

install_addon komut satırı seçeneği
--addon, 392
--configuration, 392
--update, 392

INTERLEDGER_PAYMENT_POINTERS
setting, 354

iOS
translation, 85

IP_BEHIND_REVERSE_PROXY
setting, 354

IP_PROXY_HEADER
setting, 354

IP_PROXY_OFFSET
setting, 355

J

Java properties
file format, 81

Joomla translations
file format, 83

JSON
file format, 86

L

LEGAL_TOS_DATE
setting, 355

LEGAL_URL
setting, 355

LICENSE_EXTRA
setting, 356

LICENSE_FILTER
setting, 356

LICENSE_REQUIRED
setting, 356

LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
setting, 356

list_languages
weblate admin command, 392

list_translators
weblate admin command, 392

list_translators komut satırı seçeneği
--language-code, 392

list_versions
weblate admin command, 393

list-components
wlc komut satırı seçeneği, 155

list-languages
wlc komut satırı seçeneği, 155

list-projects
wlc komut satırı seçeneği, 155

list-translations

wlc komut satırı seçeneği, 155
load() (*wlc.config.WeblateConfig* yöntemi), 160

loadpo
weblate admin command, 393

loadpo komut satırı seçeneği
--force, 393
--lang, 393

LOCALIZE_CDN_PATH
setting, 357

LOCALIZE_CDN_URL
setting, 357

lock
wlc komut satırı seçeneği, 156

lock_translation
weblate admin command, 393

lock-status
wlc komut satırı seçeneği, 156

LOGIN_REQUIRED_URLS
setting, 357

LOGIN_REQUIRED_URLS_EXCEPTIONS
setting, 357

ls
wlc komut satırı seçeneği, 155

M

main() (*wlc.main* modülü içinde), 160

MATOMO_SITE_ID
setting, 358

MATOMO_URL
setting, 358

mi18n lang
file format, 82

modül
wlc, 159
wlc.config, 160
wlc.main, 160

monolingual
translation, 75

move_language
weblate admin command, 393

N

NEARBY_MESSAGES
setting, 358

O

ortam değişkeni
CELERY_BACKUP_OPTIONS, 166, 184
CELERY_BEAT_OPTIONS, 166, 184
CELERY_MAIN_OPTIONS, 166, 184
CELERY_MEMORY_OPTIONS, 166, 184
CELERY_NOTIFY_OPTIONS, 166, 184
CELERY_TRANSLATE_OPTIONS, 166, 184
POSTGRES_ALTER_ROLE, 179
POSTGRES_CONN_MAX_AGE, 179
POSTGRES_DATABASE, 179
POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 179

POSTGRES_HOST, 179
 POSTGRES_PASSWORD, 178
 POSTGRES_PASSWORD_FILE, 178
 POSTGRES_PORT, 179
 POSTGRES_SSL_MODE, 179
 POSTGRES_USER, 179
 REDIS_DB, 180
 REDIS_HOST, 180
 REDIS_PASSWORD, 180
 REDIS_PASSWORD_FILE, 180
 REDIS_PORT, 180
 REDIS_TLS, 180
 REDIS_VERIFY_SSL, 180
 ROLLBAR_ENVIRONMENT, 182
 ROLLBAR_KEY, 182
 SENTRY_DSN, 182
 SENTRY_ENVIRONMENT, 182
 SOCIAL_AUTH_SLACK_SECRET, 178
 WEB_WORKERS, 166, 184
 WEBLATE_ADD_ADDONS, 183
 WEBLATE_ADD_APPS, 183
 WEBLATE_ADD_AUTOFIX, 183
 WEBLATE_ADD_CHECK, 183
 WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTED, 170
 WEBLATE_ADMIN_EMAIL, 166--168
 WEBLATE_ADMIN_NAME, 166--168
 WEBLATE_ADMIN_PASSWORD, 162, 166--168
 WEBLATE_ADMIN_PASSWORD_FILE, 168
 WEBLATE_AKISMET_API_KEY, 171, 402
 WEBLATE_ALLOWED_HOSTS, 168, 211, 215, 216, 364
 WEBLATE_API_RATELIMIT_ANON, 108, 172
 WEBLATE_API_RATELIMIT_USER, 108, 173
 WEBLATE_AUTH_LDAP_BIND_DN, 174
 WEBLATE_AUTH_LDAP_BIND_PASSWORD, 174
 WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE, 174
 WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REQUIRES, 174
 WEBLATE_AUTH_LDAP_SERVER_URI, 174
 WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 174
 WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 174
 WEBLATE_AUTH_LDAP_USER_SEARCH, 174
 WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 174
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 174
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 174
 WEBLATE_AUTO_UPDATE, 182
 WEBLATE_AVATAR_URL_PREFIX, 173
 WEBLATE_BASIC_LANGUAGES, 172
 WEBLATE_BITBUCKETSERVER_HOST, 171
 WEBLATE_BITBUCKETSERVER_TOKEN, 171
 WEBLATE_BITBUCKETSERVER_USERNAME, 171
 WEBLATE_BORG_EXTRA_ARGS, 173
 WEBLATE_CONTACT_FORM, 168
 WEBLATE_CORS_ALLOWED_ORIGINS, 173
 WEBLATE_CSP_CONNECT_SRC, 172
 WEBLATE_CSP_FONT_SRC, 172
 WEBLATE_CSP_IMG_SRC, 172
 WEBLATE_CSP_SCRIPT_SRC, 172
 WEBLATE_CSP_STYLE_SRC, 172
 WEBLATE_DATABASE_BACKUP, 180
 WEBLATE_DEBUG, 167, 235
 WEBLATE_DEFAULT_ACCESS_CONTROL, 171
 WEBLATE_DEFAULT_AUTO_WATCH, 172
 WEBLATE_DEFAULT_COMMITER_EMAIL, 171
 WEBLATE_DEFAULT_COMMITER_NAME, 171
 WEBLATE_DEFAULT_FROM_EMAIL, 168
 WEBLATE_DEFAULT_PULL_MESSAGE, 171
 WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 171
 WEBLATE_DEFAULT_SHARED_TM, 171
 WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 171
 WEBLATE_EMAIL_BACKEND, 181
 WEBLATE_EMAIL_HOST, 181
 WEBLATE_EMAIL_HOST_PASSWORD, 181
 WEBLATE_EMAIL_HOST_PASSWORD_FILE, 181
 WEBLATE_EMAIL_HOST_USER, 181
 WEBLATE_EMAIL_PORT, 181
 WEBLATE_EMAIL_USE_SSL, 181
 WEBLATE_EMAIL_USE_TLS, 181
 WEBLATE_ENABLE_AVATARS, 173
 WEBLATE_ENABLE_HOOKS, 173
 WEBLATE_ENABLE_HTTPS, 169, 245
 WEBLATE_ENABLE_SHARING, 173
 WEBLATE_EXTRA_HTML_HEAD, 173
 WEBLATE_GET_HELP_URL, 182
 WEBLATE_GITEA_HOST, 171
 WEBLATE_GITEA_TOKEN, 170
 WEBLATE_GITEA_USERNAME, 170
 WEBLATE_GITHUB_HOST, 170, 234
 WEBLATE_GITHUB_TOKEN, 170
 WEBLATE_GITHUB_USERNAME, 170
 WEBLATE_GITLAB_HOST, 170
 WEBLATE_GITLAB_TOKEN, 170
 WEBLATE_GITLAB_USERNAME, 170
 WEBLATE_GOOGLE_ANALYTICS_ID, 170
 WEBLATE_GPG_IDENTITY, 171
 WEBLATE_HIDE_VERSION, 172
 WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 169
 WEBLATE_IP_PROXY_HEADER, 169
 WEBLATE_LEGAL_URL, 182
 WEBLATE_LICENSE_FILTER, 172
 WEBLATE_LICENSE_REQUIRED, 172
 WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 173

WEBLATE_LOCALIZE_CDN_PATH, 183	176
WEBLATE_LOCALIZE_CDN_URL, 183	WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET,
WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS,	176
170	WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY,
WEBLATE_LOGLEVEL, 167	176
WEBLATE_LOGLEVEL_DATABASE, 167	WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET,
WEBLATE_NO_EMAIL_AUTH, 178	176
WEBLATE_PAGURE_HOST, 171	WEBLATE_SOCIAL_AUTH_FEDORA, 177
WEBLATE_PAGURE_TOKEN, 171	WEBLATE_SOCIAL_AUTH_GITEA_API_URL,
WEBLATE_PAGURE_USERNAME, 171	176
WEBLATE_PRIVACY_URL, 182	WEBLATE_SOCIAL_AUTH_GITEA_KEY, 176
WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN,	WEBLATE_SOCIAL_AUTH_GITEA_SECRET,
173	176
WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE,	WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 175
173	WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY,
WEBLATE_RATELIMIT_ATTEMPTS, 172, 404	175
WEBLATE_RATELIMIT_LOCKOUT, 172	WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME,
WEBLATE_RATELIMIT_WINDOW, 172	175
WEBLATE_REGISTRATION_ALLOW_BACKENDS,	WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET,
168	175
WEBLATE_REGISTRATION_OPEN, 168	WEBLATE_SOCIAL_AUTH_GITHUB_SECRET,
WEBLATE_REGISTRATION_REBIND, 168	175
WEBLATE_REMOVE_ADDONS, 183	WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID,
WEBLATE_REMOVE_APPS, 183	175
WEBLATE_REMOVE_AUTOFIX, 183	WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY,
WEBLATE_REMOVE_CHECK, 183	175
WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS,	WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET,
170	175
WEBLATE_REQUIRE_LOGIN, 170, 363	WEBLATE_SOCIAL_AUTH_GITLAB_API_URL,
WEBLATE_SAML_IDP_ENTITY_ID, 178	176
WEBLATE_SAML_IDP_IMAGE, 178	WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 176
WEBLATE_SAML_IDP_TITLE, 178	WEBLATE_SOCIAL_AUTH_GITLAB_SECRET,
WEBLATE_SAML_IDP_URL, 178	176
WEBLATE_SAML_IDP_X509CERT, 178	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY,
WEBLATE_SECURE_PROXY_SSL_HEADER,	176
169, 170	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET,
WEBLATE_SERVER_EMAIL, 168	176
WEBLATE_SERVICE, 166, 184	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED,
WEBLATE_SILENCED_SYSTEM_CHECKS,	176
172, 242	WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED,
WEBLATE_SIMPLIFY_LANGUAGES, 171	176
WEBLATE_SITE_DOMAIN, 167, 213, 231, 364	WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL,
WEBLATE_SITE_TITLE, 167	177
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY,	WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM,
177	177
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET,	WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URI,
177	177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY,	WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE,
177	177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET,	WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY,
177	177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_PUBLIC_KEY,	WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY,
177	177
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY,	WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET,
176	177
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY,	WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE,
176	177
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET,	WEBLATE_SOCIAL_AUTH_OIDC_KEY, 178

- WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT, [178](#)
 - WEBLATE_SOCIAL_AUTH_OIDC_SECRET, [178](#)
 - WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY, [178](#)
 - WEBLATE_SOCIAL_AUTH_OPENSUSE, [177](#)
 - WEBLATE_SOCIAL_AUTH_SLACK_KEY, [178](#)
 - WEBLATE_SOCIAL_AUTH_UBUNTU, [177](#)
 - WEBLATE_SSH_EXTRA_ARGS, [173](#)
 - WEBLATE_STATUS_URL, [182](#)
 - WEBLATE_TIME_ZONE, [169](#)
 - WEBLATE_URL_PREFIX, [172](#)
 - WEBLATE_WEBSITE_REQUIRED, [172](#)
 - WEBLATE_WORKERS, [166](#), [184](#)
 - WL_BRANCH, [339](#)
 - WL_COMPONENT_NAME, [339](#)
 - WL_COMPONENT_SLUG, [339](#)
 - WL_COMPONENT_URL, [339](#)
 - WL_ENGAGE_URL, [339](#)
 - WL_FILE_FORMAT, [339](#)
 - WL_FILEMASK, [339](#)
 - WL_LANGUAGE, [339](#)
 - WL_NEW_BASE, [339](#)
 - WL_PATH, [339](#)
 - WL_PREVIOUS_HEAD, [339](#)
 - WL_PROJECT_NAME, [339](#)
 - WL_PROJECT_SLUG, [339](#)
 - WL_REPO, [339](#)
 - WL_TEMPLATE, [339](#)
 - WL_VCS, [338](#)
- ## P
- PAGURE_CREDENTIALS
 - setting, [358](#)
 - PHP strings
 - file format, [85](#)
 - PIWIK_SITE_ID
 - setting, [358](#)
 - PIWIK_URL
 - setting, [358](#)
 - PO
 - file format, [78](#)
 - post() (*wlc. Weblate yöntemi*), [159](#)
 - post_add() (*weblate.addons.base.BaseAddon yöntemi*), [430](#)
 - post_commit() (*weblate.addons.base.BaseAddon yöntemi*), [430](#)
 - post_push() (*weblate.addons.base.BaseAddon yöntemi*), [430](#)
 - post_update() (*weblate.addons.base.BaseAddon yöntemi*), [430](#)
 - pre_commit() (*weblate.addons.base.BaseAddon yöntemi*), [431](#)
 - pre_push() (*weblate.addons.base.BaseAddon yöntemi*), [431](#)
 - pre_update() (*weblate.addons.base.BaseAddon yöntemi*), [431](#)
- ## Q
- Qt
 - file format, [83](#)
- ## R
- RATELIMIT_ATTEMPTS
 - setting, [360](#)
 - RATELIMIT_LOCKOUT
 - setting, [361](#)
 - RATELIMIT_WINDOW
 - setting, [360](#)
 - RC
 - file format, [93](#)
 - REDIS_PASSWORD, [180](#)
 - register_command() (*wlc.main modülü içinde*), [160](#)
 - REGISTRATION_ALLOW_BACKENDS
 - setting, [361](#)
 - REGISTRATION_CAPTCHA
 - setting, [361](#)
 - REGISTRATION_EMAIL_MATCH
 - setting, [361](#)
 - REGISTRATION_OPEN
 - setting, [362](#)
 - REGISTRATION_REBIND
 - setting, [362](#)
 - repo
 - wlc komut satırı seçeneği, [156](#)
 - REPOSITORY_ALERT_THRESHOLD
 - setting, [362](#)
 - REQUIRE_LOGIN
 - setting, [362](#)
- ## WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT, [178](#)
- ## WEBLATE_SOCIAL_AUTH_OIDC_SECRET, [178](#)
- ## WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY, [178](#)
- ## WEBLATE_SOCIAL_AUTH_OPENSUSE, [177](#)
- ## WEBLATE_SOCIAL_AUTH_SLACK_KEY, [178](#)
- ## WEBLATE_SOCIAL_AUTH_UBUNTU, [177](#)
- ## WEBLATE_SSH_EXTRA_ARGS, [173](#)
- ## WEBLATE_STATUS_URL, [182](#)
- ## WEBLATE_TIME_ZONE, [169](#)
- ## WEBLATE_URL_PREFIX, [172](#)
- ## WEBLATE_WEBSITE_REQUIRED, [172](#)
- ## WEBLATE_WORKERS, [166](#), [184](#)
- ## WL_BRANCH, [339](#)
- ## WL_COMPONENT_NAME, [339](#)
- ## WL_COMPONENT_SLUG, [339](#)
- ## WL_COMPONENT_URL, [339](#)
- ## WL_ENGAGE_URL, [339](#)
- ## WL_FILE_FORMAT, [339](#)
- ## WL_FILEMASK, [339](#)
- ## WL_LANGUAGE, [339](#)
- ## WL_NEW_BASE, [339](#)
- ## WL_PATH, [339](#)
- ## WL_PREVIOUS_HEAD, [339](#)
- ## WL_PROJECT_NAME, [339](#)
- ## WL_PROJECT_SLUG, [339](#)
- ## WL_REPO, [339](#)
- ## WL_TEMPLATE, [339](#)
- ## WL_VCS, [338](#)
- ## PROJECT_BACKUP_KEEP_COUNT
- ## PROJECT_BACKUP_KEEP_DAYS
- ## PROJECT_NAME_RESTRICT_RE
- ## PROJECT_WEB_RESTRICT_RE
- ## pull
- ## push
- ## pushgit
- ## Python, [158](#)
- ## Python Geliştirme Önerileri
- ## PEP 484, [426](#), [441](#)
- ## setting, [359](#)
- ## PRIVATE_COMMIT_EMAIL_OPT_IN
- ## PRIVATE_COMMIT_EMAIL_TEMPLATE
- ## setting, [359](#)
- ## setting, [359](#)
- ## setting, [359](#)
- ## setting, [360](#)
- ## setting, [360](#)
- ## setting, [360](#)
- ## setting, [360](#)
- ## setting, [360](#)
- ## wlc komut satırı seçeneği, [156](#)
- ## wlc komut satırı seçeneği, [156](#)
- ## weblate admin command, [394](#)
- ## --force-commit, [394](#)
- ## file format, [83](#)
- ## file format, [93](#)
- ## setting, [361](#)
- ## setting, [361](#)
- ## setting, [361](#)
- ## setting, [362](#)
- ## setting, [362](#)
- ## setting, [362](#)
- ## wlc komut satırı seçeneği, [156](#)
- ## setting, [362](#)
- ## setting, [362](#)

reset
 wlc komut satırı seçeneği, 156
 ResourceDictionary
 file format, 90
 REST, 105
 RESX
 file format, 90
 RFC
 RFC 5646, 75
 Ruby YAML
 file format, 92
 Ruby YAML Ain't Markup Language
 file format, 92

S

save_state() (weblate.addons.base.BaseAddon yöntemi), 431
 SECRET_KEY
 setting, 212
 SENTRY_DSN
 setting, 363
 SERVER_EMAIL
 setting, 212
 SESSION_COOKIE_AGE_AUTHENTICATED
 setting, 363
 SESSION_ENGINE
 setting, 211
 setting
 ADMINS, 211
 AKISMET_API_KEY, 342
 ALLOWED_HOSTS, 211
 ANONYMOUS_USER_NAME, 342
 AUDITLOG_EXPIRY, 342
 AUTH_LOCK_ATTEMPTS, 342
 AUTH_TOKEN_VALID, 343
 AUTO_UPDATE, 343
 AUTOFIX_LIST, 344
 AVATAR_URL_PREFIX, 343
 BACKGROUND_TASKS, 344
 BASIC_LANGUAGES, 345
 BITBUCKETSERVER_CREDENTIALS, 353
 BORG_EXTRA_ARGS, 345
 CACHE_DIR, 345
 CHECK_LIST, 346
 COMMENT_CLEANUP_DAYS, 346
 COMMIT_PENDING_HOURS, 347
 CONTACT_FORM, 347
 CSP_CONNECT_SRC, 346
 CSP_FONT_SRC, 346
 CSP_IMG_SRC, 346
 CSP_SCRIPT_SRC, 346
 CSP_STYLE_SRC, 346
 DATA_DIR, 347
 DATABASE_BACKUP, 348
 DATABASES, 211
 DEBUG, 211
 DEFAULT_ACCESS_CONTROL, 348
 DEFAULT_ADD_MESSAGE, 349
 DEFAULT_ADDON_MESSAGE, 349
 DEFAULT_ADDONS, 349
 DEFAULT_AUTO_WATCH, 348
 DEFAULT_COMMIT_MESSAGE, 349
 DEFAULT_COMMITTER_EMAIL, 349
 DEFAULT_COMMITTER_NAME, 350
 DEFAULT_DELETE_MESSAGE, 349
 DEFAULT_FROM_EMAIL, 211
 DEFAULT_LANGUAGE, 350
 DEFAULT_MERGE_MESSAGE, 349
 DEFAULT_MERGE_STYLE, 350
 DEFAULT_PAGE_LIMIT, 358
 DEFAULT_PULL_MESSAGE, 351
 DEFAULT_RESTRICTED_COMPONENT, 349
 DEFAULT_SHARED_TM, 350
 DEFAULT_TRANSLATION_PROPAGATION, 350
 ENABLE_AVATARS, 351
 ENABLE_HOOKS, 351
 ENABLE_HTTPS, 351
 ENABLE_SHARING, 351
 EXTRA_HTML_HEAD, 352
 GET_HELP_URL, 352
 GITEA_CREDENTIALS, 352
 GITHUB_CREDENTIALS, 353
 GITLAB_CREDENTIALS, 352
 GOOGLE_ANALYTICS_ID, 353
 HIDE_REPO_CREDENTIALS, 353
 HIDE_VERSION, 354
 INTERLEDGER_PAYMENT_POINTERS, 354
 IP_BEHIND_REVERSE_PROXY, 354
 IP_PROXY_HEADER, 354
 IP_PROXY_OFFSET, 355
 LEGAL_TOS_DATE, 355
 LEGAL_URL, 355
 LICENSE_EXTRA, 356
 LICENSE_FILTER, 356
 LICENSE_REQUIRED, 356
 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 356
 LOCALIZE_CDN_PATH, 357
 LOCALIZE_CDN_URL, 357
 LOGIN_REQUIRED_URLS, 357
 LOGIN_REQUIRED_URLS_EXCEPTIONS, 357
 MATOMO_SITE_ID, 358
 MATOMO_URL, 358
 NEARBY_MESSAGES, 358
 PAGURE_CREDENTIALS, 358
 PIWIK_SITE_ID, 358
 PIWIK_URL, 358
 PRIVACY_URL, 359
 PRIVATE_COMMIT_EMAIL_OPT_IN, 359
 PRIVATE_COMMIT_EMAIL_TEMPLATE, 359
 PROJECT_BACKUP_KEEP_COUNT, 359
 PROJECT_BACKUP_KEEP_DAYS, 360
 PROJECT_NAME_RESTRICT_RE, 360
 PROJECT_WEB_RESTRICT_RE, 360
 RATELIMIT_ATTEMPTS, 360

RATELIMIT_LOCKOUT, 361
RATELIMIT_WINDOW, 360
REGISTRATION_ALLOW_BACKENDS, 361
REGISTRATION_CAPTCHA, 361
REGISTRATION_EMAIL_MATCH, 361
REGISTRATION_OPEN, 362
REGISTRATION_REBIND, 362
REPOSITORY_ALERT_THRESHOLD, 362
REQUIRE_LOGIN, 362
SECRET_KEY, 212
SENTRY_DSN, 363
SERVER_EMAIL, 212
SESSION_COOKIE_AGE_AUTHENTICATED, 363
SESSION_ENGINE, 211
SIMPLIFY_LANGUAGES, 363
SINGLE_PROJECT, 364
SITE_DOMAIN, 363
SITE_TITLE, 364
SPECIAL_CHARS, 364
SSH_EXTRA_ARGS, 364
STATUS_URL, 364
SUGGESTION_CLEANUP_DAYS, 365
UPDATE_LANGUAGES, 365
URL_PREFIX, 365
VCS_API_DELAY, 365
VCS_BACKENDS, 366
VCS_CLONE_DEPTH, 366
WEBLATE_ADDONS, 366
WEBLATE_EXPORTERS, 367
WEBLATE_FORMATS, 367
WEBLATE_GPG_IDENTITY, 368
WEBLATE_MACHINERY, 368
WEBSITE_REQUIRED, 368
setupgroups
 weblate admin command, 394
setupgroups komut satırı seçeneği
 --no-privs-update, 394
 --no-projects-update, 394
setuplang
 weblate admin command, 395
setuplang komut satırı seçeneği
 --no-update, 395
show
 wlc komut satırı seçeneği, 155
SIMPLIFY_LANGUAGES
 setting, 363
SINGLE_PROJECT
 setting, 364
SITE_DOMAIN
 setting, 363
SITE_TITLE
 setting, 364
SPECIAL_CHARS
 setting, 364
SSH_EXTRA_ARGS
 setting, 364
stats

 wlc komut satırı seçeneği, 156
STATUS_URL
 setting, 364
store_post_load() (weblate.addons.base.BaseAddon yöntemi), 431
string resources
 file format, 84
SUGGESTION_CLEANUP_DAYS
 setting, 365

T

translation
 bilingual, 75
 iOS, 85
 monolingual, 75
TS
 file format, 83

U

unit_pre_create() (weblate.addons.base.BaseAddon yöntemi), 431
unlock
 wlc komut satırı seçeneği, 156
unlock_translation
 weblate admin command, 394
UPDATE_LANGUAGES
 setting, 365
updatechecks
 weblate admin command, 395
updategit
 weblate admin command, 395
upload
 wlc komut satırı seçeneği, 156
URL_PREFIX
 setting, 365

V

VCS_API_DELAY
 setting, 365
VCS_BACKENDS
 setting, 366
VCS_CLONE_DEPTH
 setting, 366
version
 wlc komut satırı seçeneği, 155

W

WEB_WORKERS, 166, 184
Weblate (wlc içindeki sınıf), 159
weblate admin command
 add_suggestions, 385
 auto_translate, 385
 celery_queues, 386
 checkgit, 386
 cleanup_ssh_keys, 387
 cleanuptrans, 387
 commit_pending, 386
 commitgit, 386

- createadmin, 387
 - dump_memory, 388
 - dumpuserdata, 388
 - import_demo, 388
 - import_json, 388
 - import_memory, 389
 - import_project, 389
 - importuserdata, 391
 - importusers, 391
 - install_addon, 392
 - list_languages, 392
 - list_translators, 392
 - list_versions, 393
 - loadpo, 393
 - lock_translation, 393
 - move_language, 393
 - pushgit, 394
 - setupgroups, 394
 - setuplang, 395
 - unlock_translation, 394
 - updatechecks, 395
 - updategit, 395
 - WEBLATE_ADDONS
 - setting, 366
 - WEBLATE_ADMIN_EMAIL, 166--168
 - WEBLATE_ADMIN_NAME, 166--168
 - WEBLATE_ADMIN_PASSWORD, 162, 166--168
 - WEBLATE_ADMIN_PASSWORD_FILE, 168
 - WEBLATE_AKISMET_API_KEY, 402
 - WEBLATE_ALLOWED_HOSTS, 211, 215, 216, 364
 - WEBLATE_API_RATELIMIT_ANON, 108
 - WEBLATE_API_RATELIMIT_USER, 108
 - WEBLATE_AUTH_LDAP_BIND_PASSWORD, 174
 - WEBLATE_DEBUG, 235
 - WEBLATE_EMAIL_HOST_PASSWORD, 181
 - WEBLATE_EMAIL_PORT, 181
 - WEBLATE_EMAIL_USE_SSL, 181
 - WEBLATE_EMAIL_USE_TLS, 181
 - WEBLATE_ENABLE_HTTPS, 245
 - WEBLATE_EXPORTERS
 - setting, 367
 - WEBLATE_FORMATS
 - setting, 367
 - WEBLATE_GITHUB_HOST, 234
 - WEBLATE_GPG_IDENTITY
 - setting, 368
 - WEBLATE_LOCALIZE_CDN_PATH, 183
 - WEBLATE_MACHINERY
 - setting, 368
 - WEBLATE_RATELIMIT_ATTEMPTS, 404
 - WEBLATE_REQUIRE_LOGIN, 363
 - WEBLATE_SECURE_PROXY_SSL_HEADER, 169
 - WEBLATE_SERVICE, 166
 - WEBLATE_SILENCED_SYSTEM_CHECKS, 242
 - WEBLATE_SITE_DOMAIN, 213, 231, 364
 - WEBLATE_WORKERS, 166, 184
 - WeblateConfig (*wlc.config* içindeki sınıf), 160
 - WeblateException, 159
 - WEBSITE_REQUIRED
 - setting, 368
 - wlc, 153
 - modülü, 159
 - wlc komut satırı seçeneği
 - author-email, 157
 - author-name, 157
 - config, 155
 - config-section, 155
 - convert, 156
 - format, 155
 - fuzzy, 157
 - input, 156
 - key, 155
 - method, 157
 - output, 156
 - overwrite, 156
 - url, 155
 - changes, 156
 - cleanup, 156
 - commit, 155
 - download, 156
 - list-components, 155
 - list-languages, 155
 - list-projects, 155
 - list-translations, 155
 - lock, 156
 - lock-status, 156
 - ls, 155
 - pull, 156
 - push, 156
 - repo, 156
 - reset, 156
 - show, 155
 - stats, 156
 - unlock, 156
 - upload, 156
 - version, 155
 - wlc.config
 - modülü, 160
 - wlc.main
 - modülü, 160
 - WPF
 - file format, 90
- ## X
- XLIFF
 - file format, 79
 - XML
 - file format, 93
- ## Y
- YAML
 - file format, 91
 - YAML Ain't Markup Language
 - file format, 91