The Weblate Manual

Versión 4.18.2

Michal Čihař

30 de junio de 2023

Documentación para usuarios

1 Documentación para usuarios **1**
- 1.1 Fundamentos de Weblate . . . 1
- 1.2 Registro y perfil de usuario . . . 1
- 1.3 Traducir con Weblate . . . 11
- 1.4 Descargar y cargar traducciones . . . 23
- 1.5 Glosario . . . 27
- 1.6 Comprobaciones y correcciones . . . 29
- 1.7 Búsquedas . . . 63
- 1.8 Flujos de trabajo de traducción . . . 70
- 1.9 Preguntas frecuentes . . . 74
- 1.10 Formatos de archivo admitidos . . . 82
- 1.11 Integración de control de versiones . . . 110
- 1.12 API REST de Weblate . . . 117
- 1.13 Cliente de Weblate . . . 165
- 1.14 API de Python de Weblate . . . 170

2 Documentación para administradores **173**
- 2.1 Instrucciones de configuración . . . 173
- 2.2 Implantaciones de Weblate . . . 240
- 2.3 Actualizar Weblate . . . 241
- 2.4 Respaldar y trasladar Weblate . . . 250
- 2.5 Autenticación . . . 257
- 2.6 Control de acceso . . . 268
- 2.7 Proyectos de traducción . . . 278
- 2.8 Definiciones de idioma . . . 296
- 2.9 Regionalización continua . . . 299
- 2.10 Licensing translations . . . 307
- 2.11 Proceso de traducción . . . 309
- 2.12 Comprobaciones y correcciones . . . 315
- 2.13 Configuración de las sugerencias automáticas . . . 326
- 2.14 Complementos . . . 337
- 2.15 Memoria de traducción . . . 358
- 2.16 Configuración . . . 360
- 2.17 Configuración de muestra . . . 389
- 2.18 Órdenes de gestión . . . 405
- 2.19 Anuncios . . . 417
- 2.20 Listas de componentes . . . 419
- 2.21 Módulos opcionales de Weblate . . . 420
- 2.22 Personalizar Weblate . . . 425
- 2.23 Interfaz de gestión . . . 427
- 2.24 Obtener ayuda con Weblate . . . 436

2.25 Documentos legales . . . 439

3 Documentación para contribuidores **441**
3.1 Contribuir con Weblate . . . 441
3.2 Empezar a contribuir con código a Weblate . . . 443
3.3 Código fuente de Weblate . . . 447
3.4 Depurar Weblate . . . 448
3.5 Weblate por dentro . . . 450
3.6 Developing add-ons . . . 451
3.7 Interfaz de Weblate . . . 453
3.8 Informar de problemas en Weblate . . . 454
3.9 Weblate testsuite and continuous integration . . . 454
3.10 Esquemas de datos . . . 456
3.11 Releasing Weblate . . . 460
3.12 Seguridad y privacidad . . . 461
3.13 Contributing to Weblate modules . . . 462
3.14 Acerca de Weblate . . . 463
3.15 Licencia . . . 464

4 Historial de cambios **465**
4.1 Weblate 4.18.2 . . . 465
4.2 Weblate 4.18.1 . . . 465
4.3 Weblate 4.18 . . . 465
4.4 Weblate 4.17 . . . 466
4.5 Weblate 4.16.4 . . . 467
4.6 Weblate 4.16.3 . . . 467
4.7 Weblate 4.16.2 . . . 467
4.8 Weblate 4.16.1 . . . 467
4.9 Weblate 4.16 . . . 467
4.10 Weblate 4.15.2 . . . 468
4.11 Weblate 4.15.1 . . . 468
4.12 Weblate 4.15 . . . 468
4.13 Weblate 4.14.2 . . . 469
4.14 Weblate 4.14.1 . . . 470
4.15 Weblate 4.14 . . . 470
4.16 Weblate 4.13.1 . . . 471
4.17 Weblate 4.13 . . . 471
4.18 Weblate 4.12.2 . . . 472
4.19 Weblate 4.12.1 . . . 472
4.20 Weblate 4.12 . . . 472
4.21 Weblate 4.11.2 . . . 473
4.22 Weblate 4.11.1 . . . 473
4.23 Weblate 4.11 . . . 473
4.24 Weblate 4.10.1 . . . 474
4.25 Weblate 4.10 . . . 474
4.26 Weblate 4.9.1 . . . 475
4.27 Weblate 4.9 . . . 475
4.28 Weblate 4.8.1 . . . 476
4.29 Weblate 4.8 . . . 476
4.30 Weblate 4.7.2 . . . 477
4.31 Weblate 4.7.1 . . . 477
4.32 Weblate 4.7 . . . 477
4.33 Weblate 4.6.2 . . . 478
4.34 Weblate 4.6.1 . . . 478
4.35 Weblate 4.6 . . . 478
4.36 Weblate 4.5.3 . . . 479
4.37 Weblate 4.5.2 . . . 479
4.38 Weblate 4.5.1 . . . 480

4.39 Weblate 4.5 . . . 480
4.40 Weblate 4.4.2 . . . 481
4.41 Weblate 4.4.1 . . . 481
4.42 Weblate 4.4 . . . 481
4.43 Weblate 4.3.2 . . . 482
4.44 Weblate 4.3.1 . . . 483
4.45 Weblate 4.3 . . . 483
4.46 Weblate 4.2.2 . . . 484
4.47 Weblate 4.2.1 . . . 484
4.48 Weblate 4.2 . . . 484
4.49 Weblate 4.1.1 . . . 485
4.50 Weblate 4.1 . . . 485
4.51 Weblate 4.0.4 . . . 487
4.52 Weblate 4.0.3 . . . 487
4.53 Weblate 4.0.2 . . . 487
4.54 Weblate 4.0.1 . . . 488
4.55 Weblate 4.0 . . . 488
4.56 Serie 3.x de Weblate . . . 489
4.57 Serie 2.x de Weblate . . . 500
4.58 Serie 1.x de Weblate . . . 511
4.59 Serie 0.x de Weblate . . . 515

Índice de Módulos Python **519**

Tabla de enrutamiento HTTP **520**

Índice **523**

CAPÍTULO 1

Documentación para usuarios

1.1 Fundamentos de Weblate

1.1.1 Estructura de los proyectos y los componentes

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 *gettext de GNU* or *Recursos de cadenas de Android*). 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 *Configuración de componentes* using *Permitir propagación de traducciones* in case the translations should diverge.

Ver también:

../devel/integration

1.2 Registro y perfil de usuario

1.2.1 Registro

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:

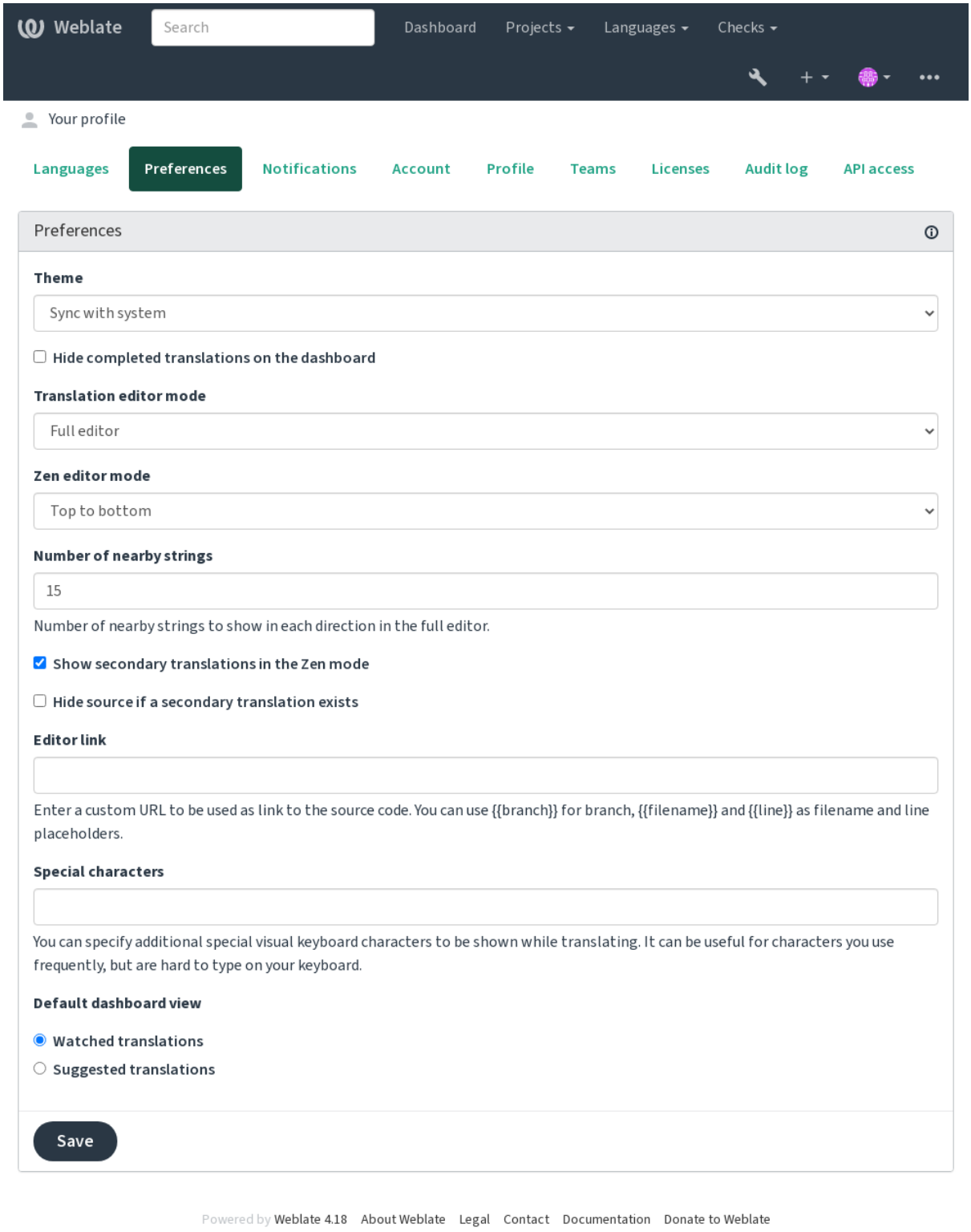
1. Fill out the registration form with your credentials.
2. Activate registration by following the link in the e-mail you receive.
3. Optionally adjust your profile to choose which languages you know.

1.2.2 Cuadro de mando

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

De manera predeterminada se muestran los componentes de los proyectos que monitoriza, así como enlaces directos en las lenguas preferidas que haya establecido.

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



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 *Listas de componentes*.

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

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

1.2.3 Perfil de usuario

Para acceder a su perfil de usuario, pulse en el icono de su cuenta en el extremo derecho del menú superior y, en el menú, elija *Configuración*.

El perfil de usuario contiene sus preferencias. El nombre y la dirección de correo electrónico se utilizan para las consignas de los sistemas de control de versiones, de modo que mantenga estos datos al día.

Nota: All language selections only offer currently translated languages.

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

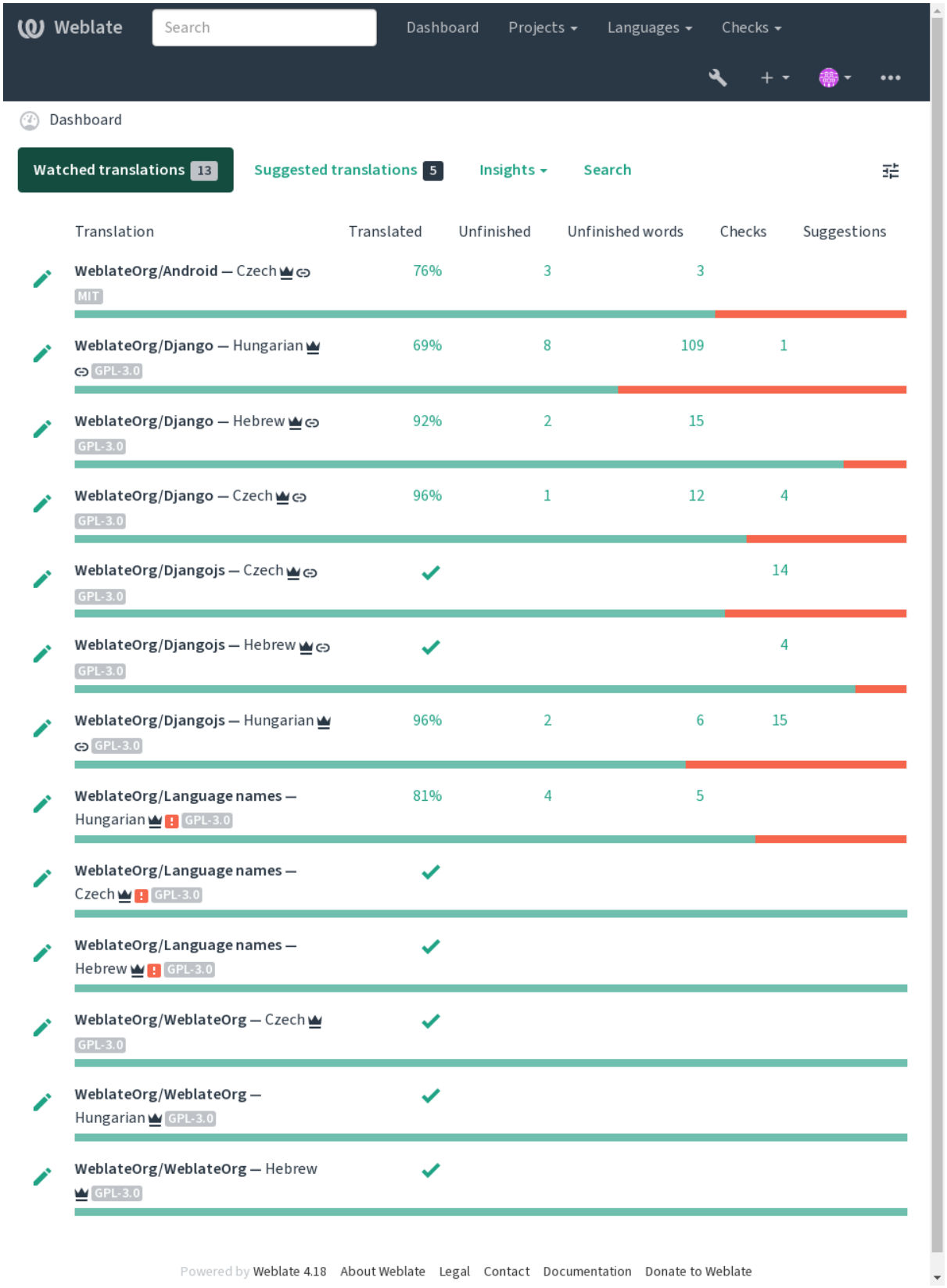
Idiomas

1.2.4 Interface language

Choose the language you want to display the UI in.

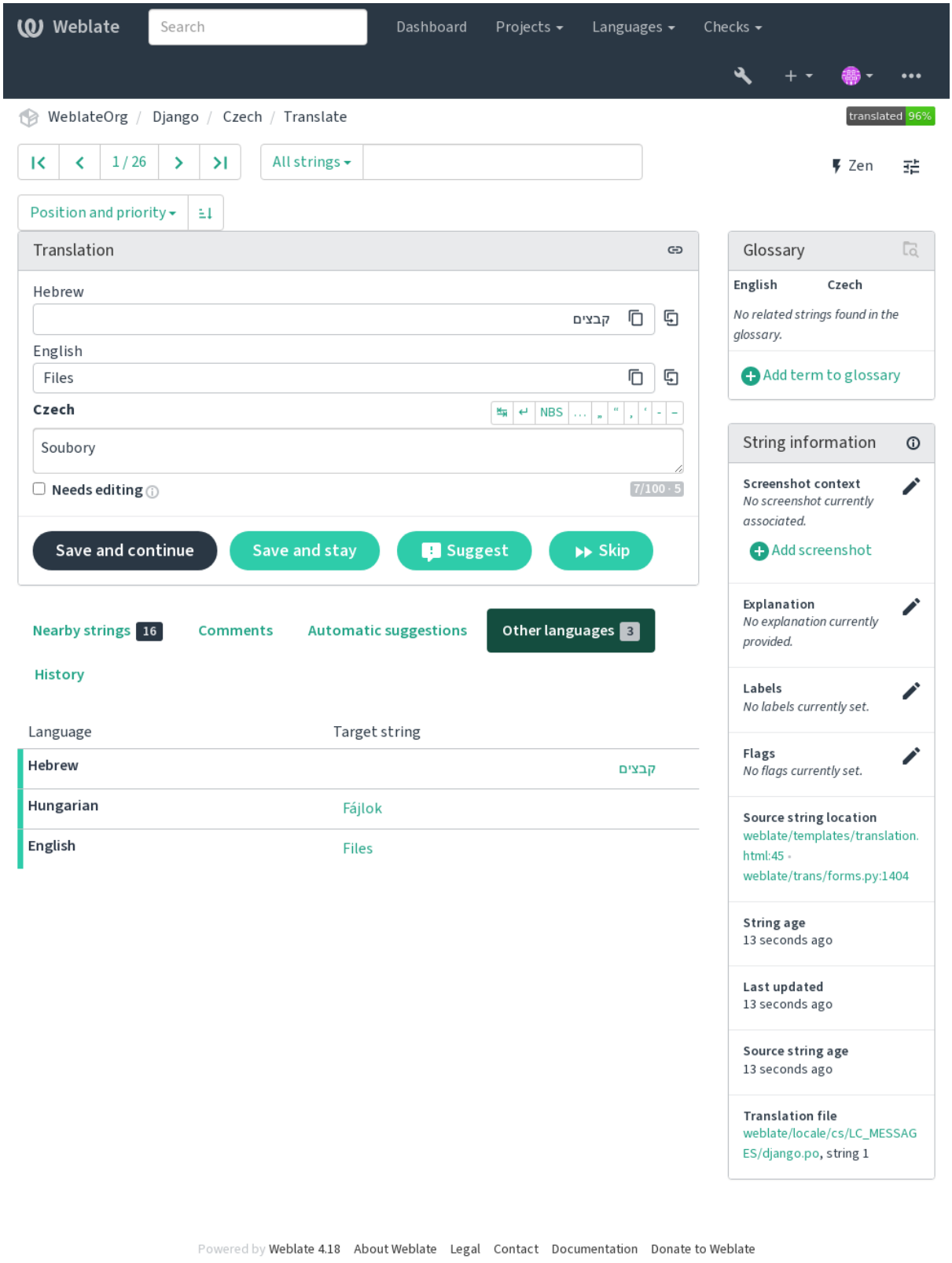
Idiomas traducidos

Elija los idiomas en los que prefiera traducir y se le ofrecerán en la página principal de los proyectos supervisados, de tal modo que se le facilite el acceso a las traducciones en esos idiomas.


Weblate
Search
Dashboard
Projects
Languages
Checks
Dashboard
Watched translations 13
Suggested translations 5
Insights
Search
Translation
Translated
Unfinished
Unfinished words
Checks
Suggestions
WeblateOrg/Android — Czech
MIT
76%
3
3
WeblateOrg/Django — Hungarian
GPL-3.0
69%
8
109
1
WeblateOrg/Django — Hebrew
GPL-3.0
92%
2
15
WeblateOrg/Django — Czech
GPL-3.0
96%
1
12
4
WeblateOrg/Djangojs — Czech
GPL-3.0
14
WeblateOrg/Djangojs — Hebrew
GPL-3.0
4
WeblateOrg/Djangojs — Hungarian
GPL-3.0
96%
2
6
15
WeblateOrg/Language names — Hungarian
GPL-3.0
81%
4
5
WeblateOrg/Language names — Czech
GPL-3.0
WeblateOrg/Language names — Hebrew
GPL-3.0
WeblateOrg/WeblateOrg — Czech
GPL-3.0
WeblateOrg/WeblateOrg — Hungarian
GPL-3.0
WeblateOrg/WeblateOrg — Hebrew
GPL-3.0
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Idiomas secundarios

Puede definir qué idiomas secundarios mostrar para que le sirvan de guía al traducir. La siguiente imagen muestra un ejemplo, en el cual la lengua hebrea se muestra como secundaria:



1.2.5 Preferencias

Theme

Choose whether Weblate follows system settings for dark or light theme, or choose one of them manually.

Vista predeterminada del cuadro de mando

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.

Ver también:

Listas de componentes

Enlace del editor

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

Consejo: By setting the *Editor link*, you use your local editor to open the VCS source code file of translated strings. You can use *Marcación de plantilla*.

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

Ver también:

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

Caracteres especiales

Additional special characters to include in the *Visual keyboard*.

1.2.6 Notificaciones

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.

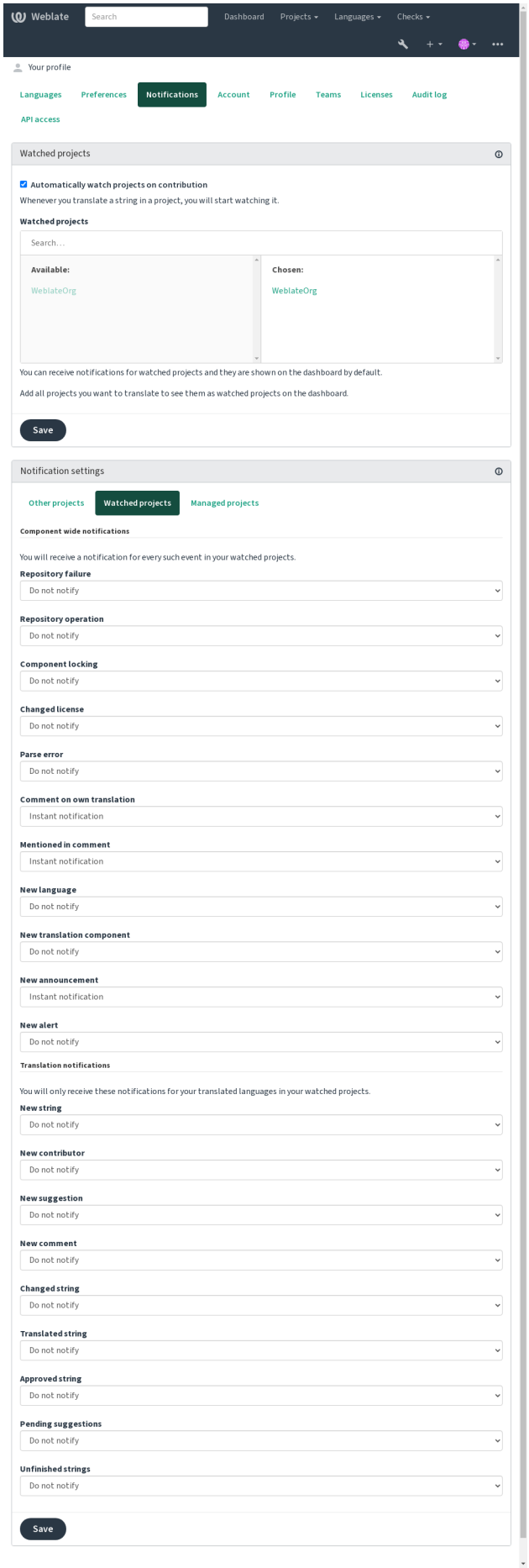
Some of the notifications are sent only for events in your languages (for example about new strings to translate), while some trigger at component level (for example merge errors). These two groups of notifications are visually separated in the settings.

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`.

Nota: No recibirá ninguna notificación por sus propias acciones.

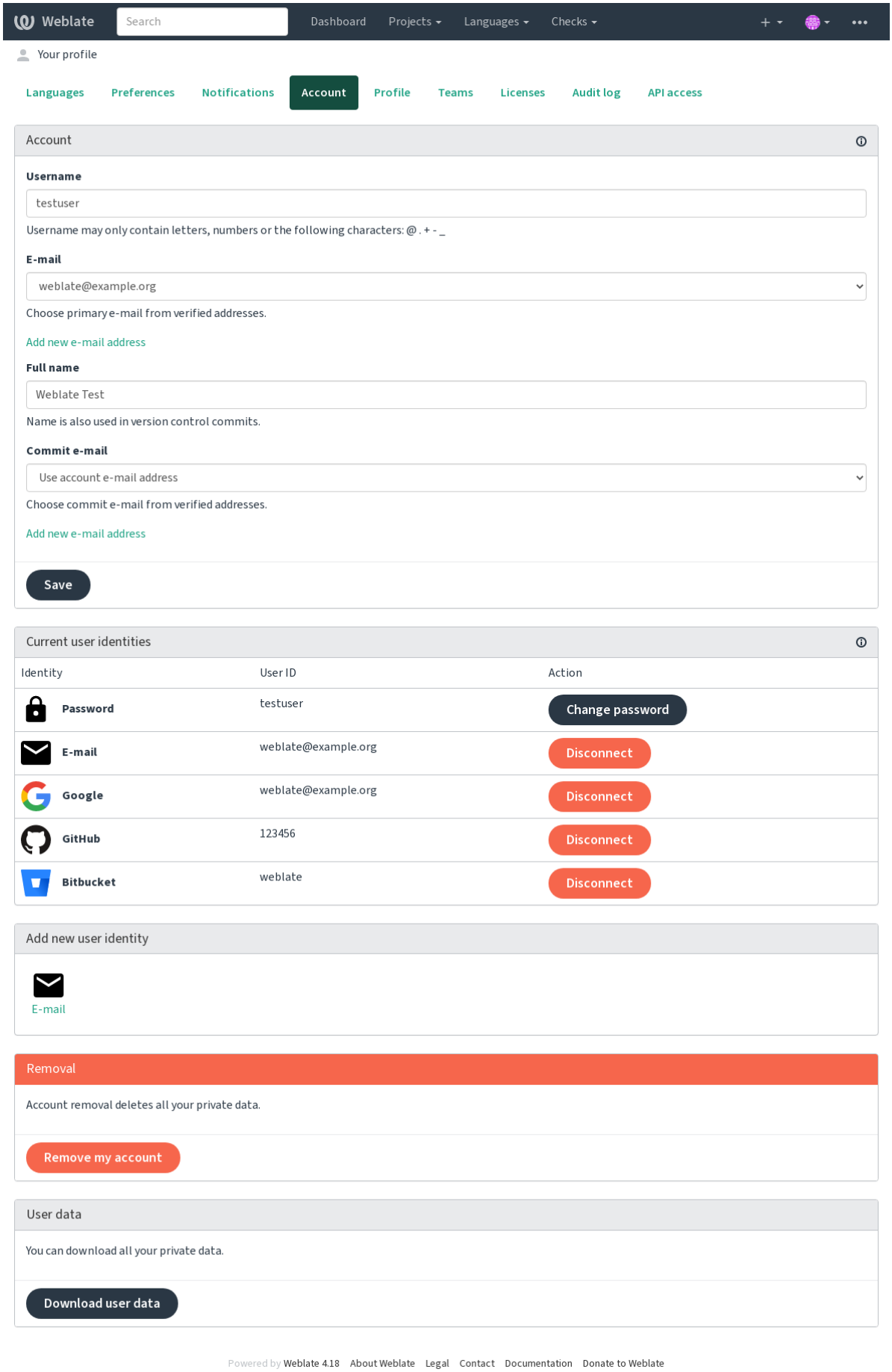
Consejo: Sending out notifications is limited, you will not receive more than 1000 e-mails per day. Any further notifications for you will be discarded.


Weblate
Search
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

1.2.7 Cuenta

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 *Exportación de datos de usuarios de Weblate*).

Nota: 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
Search
Dashboard
Projects
Languages
Checks
Your profile
Languages
Preferences
Notifications
Account
Profile
Teams
Licenses
Audit log
API access
Account
Username
testuser
Username may only contain letters, numbers or the following characters: @ . + - _
E-mail
weblate@example.org
Choose primary e-mail from verified addresses.
Add new e-mail address
Full name
Weblate Test
Name is also used in version control commits.
Commit e-mail
Use account e-mail address
Choose commit e-mail from verified addresses.
Add new e-mail address
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 user identity
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
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.2.8 Perfil

Todos los campos de esta página son opcionales, en cualquier momento puede eliminarlos y, si los rellena, consiente que compartamos esta información allá donde aparezca su perfil de usuario.

The private 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 Licencias

1.2.10 API de acceso

You can get or reset your API access token here.

1.2.11 Registro de auditoría

Audit log keeps track of the actions performed with your account. It logs IP address and browser for every important action with your account. The critical actions also trigger a notification to a primary e-mail address.

Ver también:

Running behind reverse proxy

1.3 Traducir con 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:

- El proyecto acepta traducciones directas
- The project only accepts suggestions, which are automatically validated once a defined number of votes is reached

Please see *Flujos de trabajo de traducción* for more info on translation workflow.

Opciones de visibilidad de los proyectos de traducción:

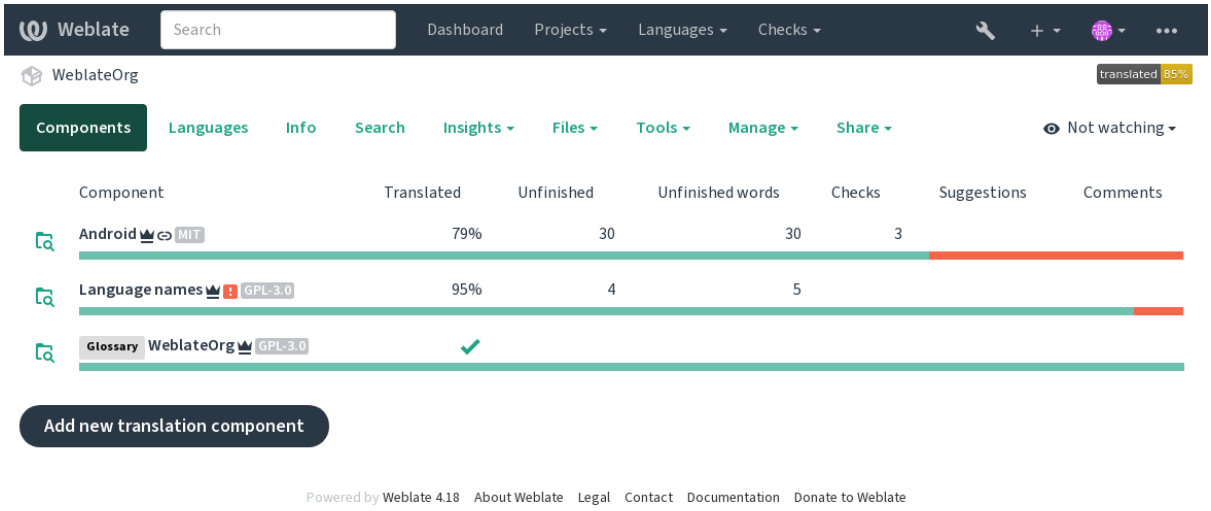
- Visible para el público
- Visible solo a un grupo determinado de traductores

Ver también:

Control de acceso, *Flujos de trabajo de traducción*

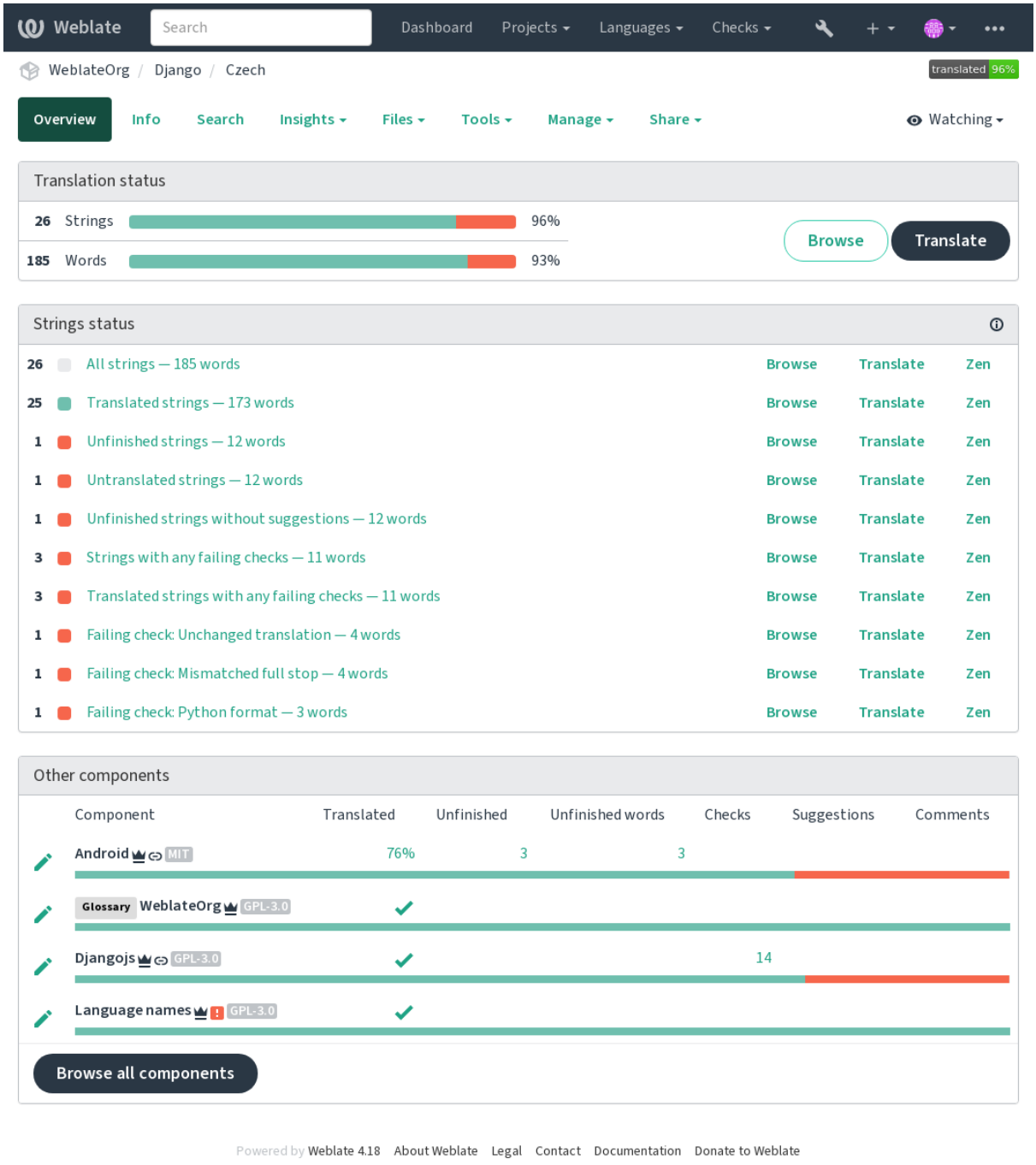
1.3.1 Proyectos de traducción

Los proyectos de traducción agrupan componentes relacionados; por ejemplo, recursos para el mismo programa, libro o proyecto.



1.3.2 Enlaces de traducción

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.



1.3.3 Sugerencias

Nota: Los permisos reales pueden variar en función de cómo se haya configurado Weblate.

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.

Las sugerencias se examinan diariamente para eliminar aquellas que son duplicados o que coinciden con la traducción aplicada actual.

1.3.4 Comentarios

Three types of comments can be posted: for translations, source strings, or to report source string bugs when this functionality is turned on using *Activar revisiones de origen*. 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.

Puede utilizar la sintaxis Markdown en todos los comentarios y mencionar a otros usuarios por `@nombredeusuario`.

Ver también:

report-source, *Source strings reviews*, *Activar revisiones de origen*

1.3.5 Variantes

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.

Ver también:

variants, *Variantes*

1.3.6 Etiquetas

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

Following labels are used by Weblate:

Traducida automáticamente
: String was translated using *Traducción automática*.

La cadena de origen necesita revisión
: La cadena fue marcada para su revisión usando revisiones de fuentes.

Ver también:

labels

1.3.7 Traducción

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.

Todos los caracteres especiales de espacio se subrayan en rojo y señalan con símbolos grises. Además, también se señala en rojo cuando hay dos espacios consecutivos para alertar al traductor de potenciales errores de formato.

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 *Idiomas secundarios*) above the source string.

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

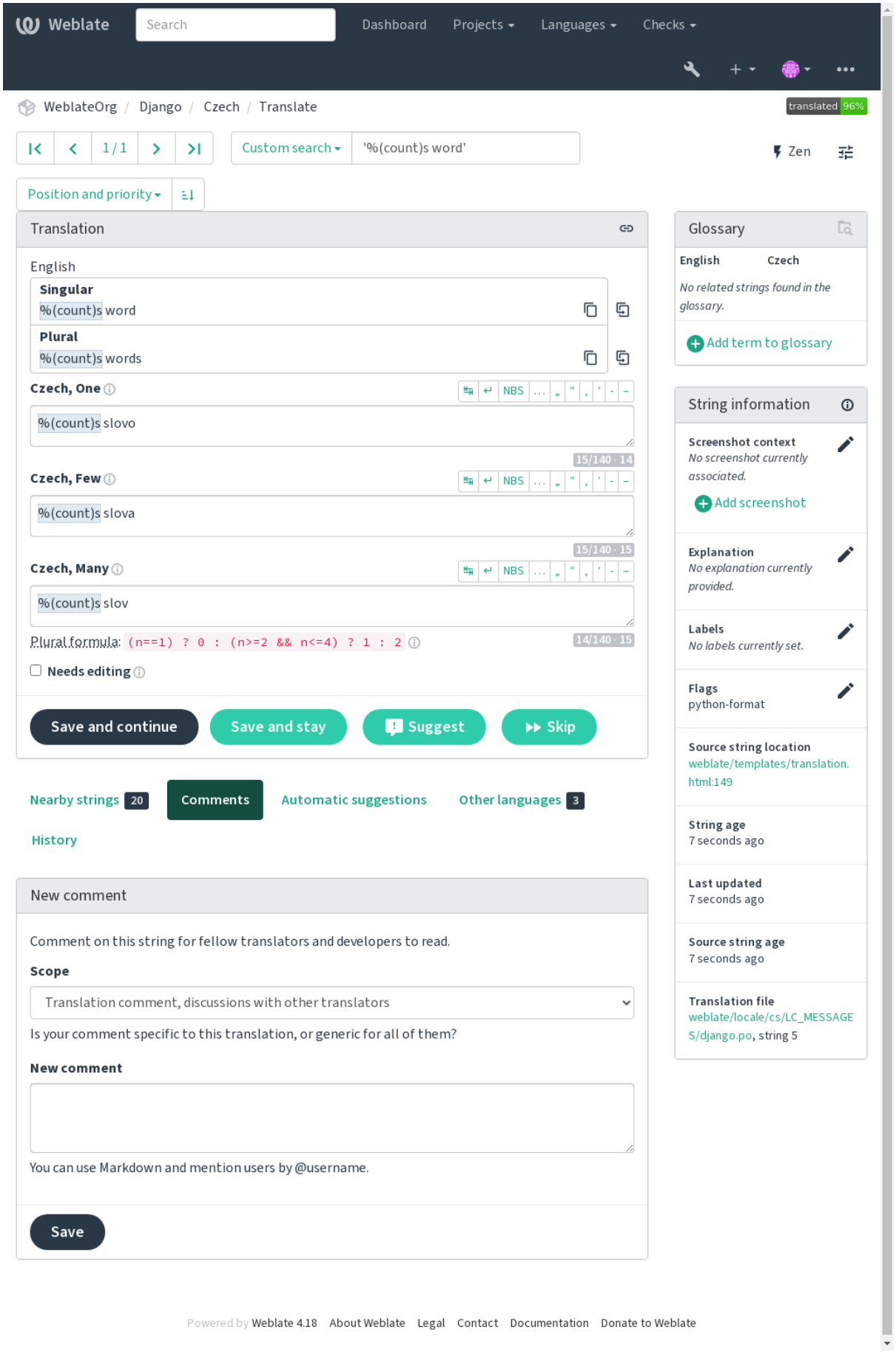
Plurales

Las palabras que cambian de forma para indicar su designación de número se denominan plurales. Cada lengua tiene su propia definición de plurales. El español, por ejemplo, admite un plural. En la definición singular de, por ejemplo, «automóvil», se hace referencia implícita a un único automóvil; en la definición plural, «automóviles», se hace referencia a dos o más automóviles, o bien, al concepto de automóviles sustantivado. En lenguas como el checo y el árabe existen más plurales; además, las reglas de pluralización que emplean también difieren.

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.

Ver también:

Fórmula de plurales


Weblate
Search
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Czech / Translate
translated 96%
1 / 1
Custom search
'%(count)s word'
Zen
Position and priority
Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
%(count)s slovo
15/140 · 14
Czech, Few
%(count)s slova
15/140 · 15
Czech, Many
%(count)s slov
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
14/140 · 15
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
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
7 seconds ago
Last updated
7 seconds ago
Source string age
7 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Alternative translations

Nuevo en la versión 4.13.

Nota: This is currently only supported with *Multivalue CSV file*.

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.

Atajos en el teclado

Los atajos de teclado siguientes pueden utilizarse durante la traducción:

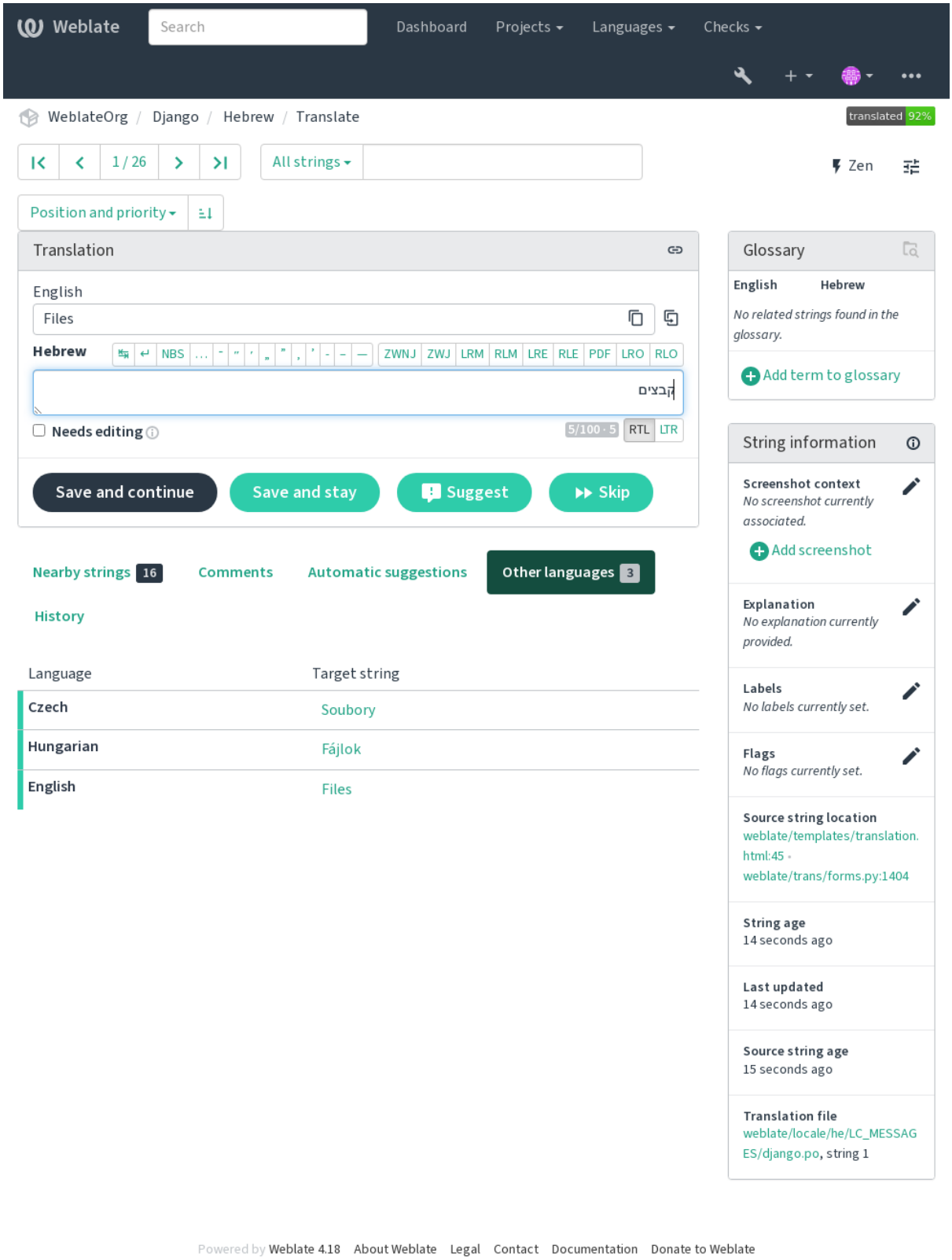
Atajo de teclado	Descripción
`Alt+Inicio`	Navegar a la primera traducción de la búsqueda actual.
`Alt+Fin`	Navegar a la última traducción de la búsqueda actual.
`Alt+Re pág` o `Ctrl+↑` or `Alt+↑` or `Cmd+↑`	Navegar a la traducción anterior de la búsqueda actual.
`Alt+Av pág` o `Ctrl+↓` o `Alt+↓` o `⌘+↓`	Navegar a la traducción siguiente de la búsqueda actual.
`Ctrl+Intro` o `⌘+Intro`	Submit current form; this is same as pressing *Save and continue* while editing translation.
`Ctrl+Mayús+Intro` o `⌘+Mayús+Intro`	Desmarque la traducción que necesita editar y envíela.
`Alt+Intro` o `Option+Enter`	Submit the string as a suggestion; this is same as pressing *Suggest* while editing translation.
`Ctrl+E` o `⌘+E`	Dar el foco al editor de traducciones.
`Ctrl+U` o `⌘+U`	Dar el foco al editor de comentarios.
`Ctrl+M` o `⌘+M`	Muestra la pestaña *Sugerencias automáticas* ; vea *Sugerencias automáticas*.
`Ctrl+1` to `Ctrl+9` or `Cmd+1` to `Cmd+9`	Copies placeable of given number from source string.
`Ctrl+M` followed by `1` to `9` or `Cmd+M` followed by `1` to `9`	Copy the machine translation of given number to current translation.
`Ctrl+I` followed by `1` to `9` or `Cmd+I` followed by `1` to `9`	Ignorar un elemento en la lista de comprobaciones fallidas.
`Ctrl+J` or `Cmd+J`	Muestra la pestaña *Cadenas cercanas*.
`Ctrl+S` or `Cmd+S`	Centrar el campo de búsqueda.
`Ctrl+O` or `Cmd+O`	Copiar la cadena de origen.
`Ctrl+Y` or `Cmd+Y`	Toggle the *Needs editing* checkbox.

Visual keyboard

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.

Los símbolos que se muestran pertenecen a una de estas tres categorías:

- User configured *Caracteres especiales* defined in the *Perfil de usuario*
- Caracteres provistos por Weblate en función del idioma (p. ej., comillas y marcas para la escritura bidireccional)
- Caracteres configurados mediante `SPECIAL_CHARS`


Weblate
Search
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Hebrew / Translate
translated 92%
1 / 26
All strings
Zen
Position and priority
Translation
English
Files
Hebrew
NBS
ZWNJ
ZWJ
LRM
RLM
LRE
RLE
PDF
LRO
RLO
קבצים
Needs editing
5/100 · 5
RTL
LTR
Save and continue
Save and stay
Suggest
Skip
Nearby strings 16
Comments
Automatic suggestions
Other languages 3
History
Language
Target string
Czech
Soubory
Hungarian
Fájlok
English
Files
Glossary
English
Hebrew
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
No flags currently set.
Source string location
weblate/templates/translation.html:45 • weblate/trans/forms.py:1404
String age
14 seconds ago
Last updated
14 seconds ago
Source string age
15 seconds ago
Translation file
weblate/locale/he/LC_MESSAGES/django.po, string 1
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Contexto de la traducción

Esta descripción contextual brinda información relativa a la cadena actual.

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

Capturas de pantalla
Screenshots can be uploaded to Weblate to better inform translators of where and how the string is used, see *Contexto visual para cadenas*.

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

Otras ocurrencias
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 *Incoherente*). You can choose which one to use.

Memoria de traducción
Look at similar strings translated in past, see *Memoria de traducción*.

Glosario
Muestra los términos del glosario del proyecto que figuran en el mensaje actual.

Cambios recientes
Lista de personas que han modificado este mensaje recientemente utilizando Weblate.

Proyecto
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.

Historial de traducciones

Todos los cambios se guardan de manera predeterminada en la base de datos y pueden revertirse (a menos que lo haya desactivado en la configuración de cada componente). Es posible asimismo revertir todo en el sistema de control de versiones subyacente.

Longitud de las cadenas traducidas

Weblate puede limitar la longitud de la traducción de varias formas para garantizar que la cadena traducida no sea demasiado extensa:

- 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 *Formatos bilingües y monolingües* for more info.
- Maximal length in characters defined by translation file or flag, see *Longitud máxima de la traducción*.
- Maximal rendered size in pixels defined by flags, see *Tamaño máximo de la traducción*.

1.3.8 Sugerencias automáticas

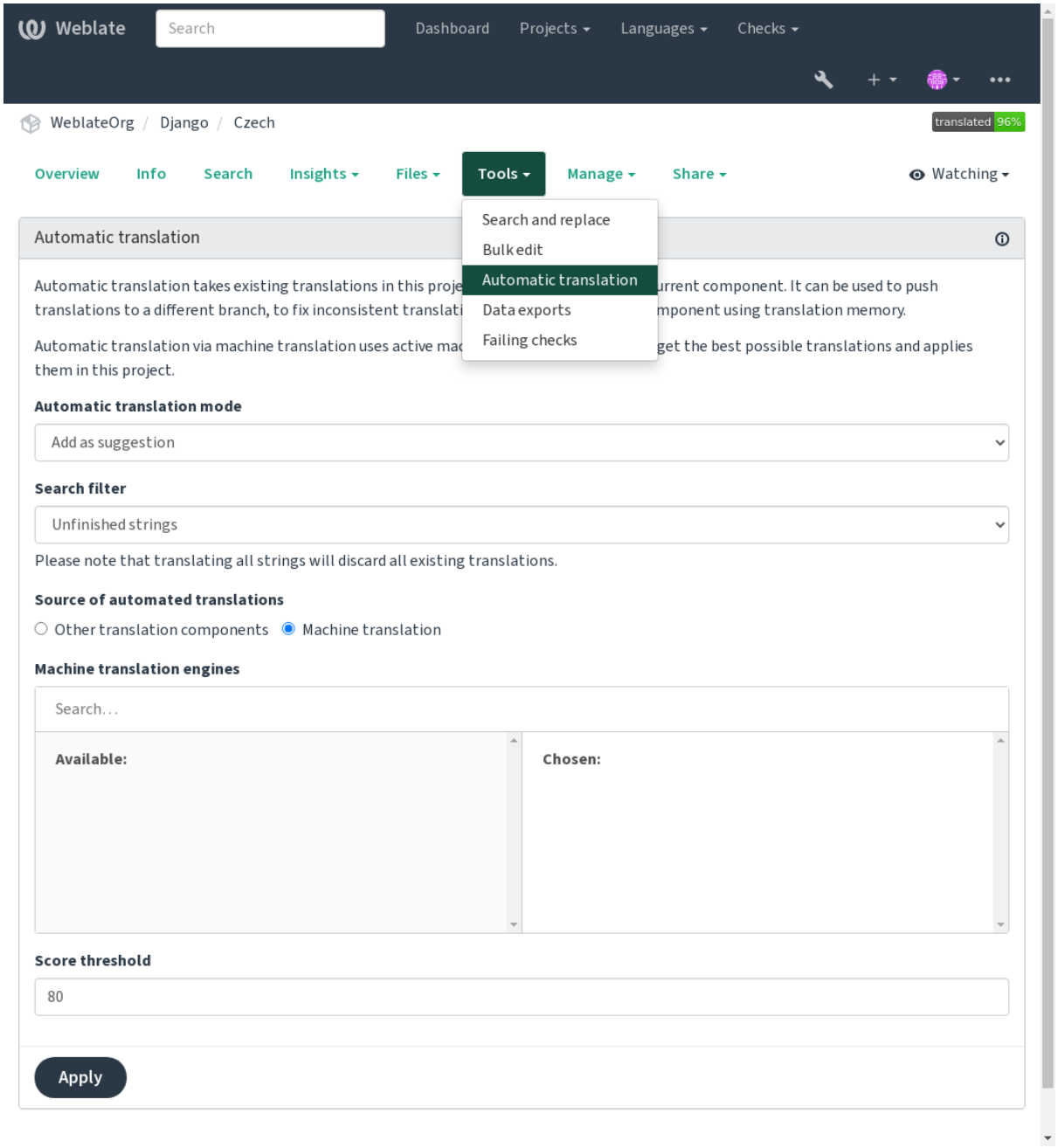
Based on configuration and your translated language, Weblate provides suggestions from several machine translation tools and *Memoria de traducción*. All machine translations are available in a single tab of each translation page.

Ver también:

You can find the list of supported tools in *Configuración de las sugerencias automáticas*.

1.3.9 Traducción automática

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



Two modes of operation are possible:

- Using other Weblate components as a source for translations.

- Using selected machine translation services with translations above a certain quality threshold.

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

Advertencia: 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).

The automatically translated strings are labelled *Automatically translated*.

Ver también:

Mantener iguales las traducciones entre los componentes

1.3.10 Rate limiting

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 *Rate limiting*.

1.3.11 Buscar y reemplazar

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

Consejo: No se preocupe de estropear las cadenas. Este proceso consta de dos etapas y obtendrá una previsualización de las cadenas modificadas antes de confirmar el cambio en sí.

1.3.12 Edición en masa

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 *Personalizar el comportamiento mediante indicadores*)
- Adjust string labels (see labels)

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

Ver también:

Bulk edit add-on

1.3.13 Matrix View

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 Mode

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 *Preferencias* tab on your *Perfil de usuario*. Here you can also choose between having translations listed *Top to bottom* or *Side by side* depending on your personal preference.

1.4 Descargar y cargar traducciones

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.

Nota: Available options might be limited by *access control* settings.

1.4.1 Descargar traducciones

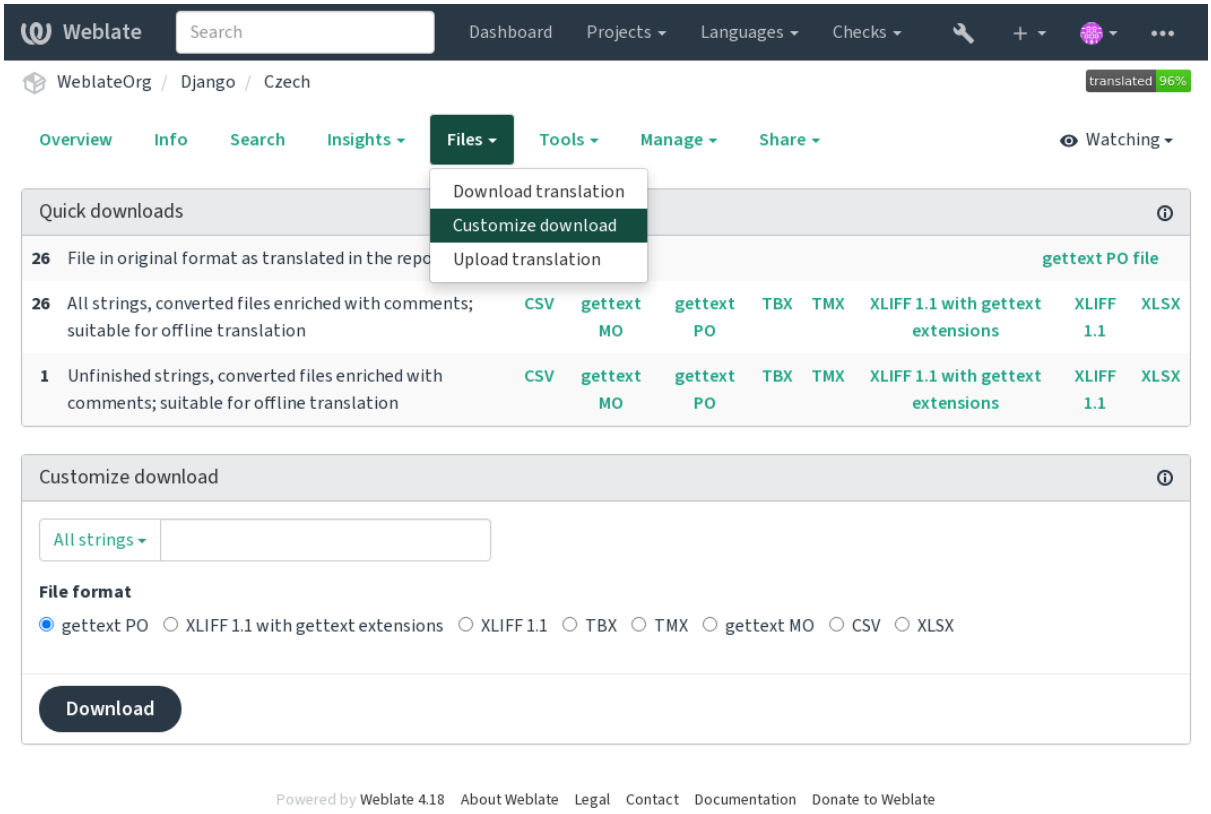
From the project or component dashboard, translatable files can be downloaded in the *Files* menu.

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 (`po`)
- XLIFF with gettext extensions (`xliff`)
- XLIFF 1.1 (`xliff11`)
- TermBase eXchange (`tbx`)
- Translation Memory eXchange (`tmx`)
- gettext MO (only available when translation is using gettext PO) (`mo`)
- CSV (`csv`)
- Excel Open XML (`xlsx`)
- JSON (only available for monolingual translations) (`json`)
- Android String Resource (only available for monolingual translations) (`aresource`)
- iOS strings (only available for monolingual translations) (`strings`)

Consejo: The content available in the converted files differs based on file format features, you can find overview in *Translation types capabilities*.

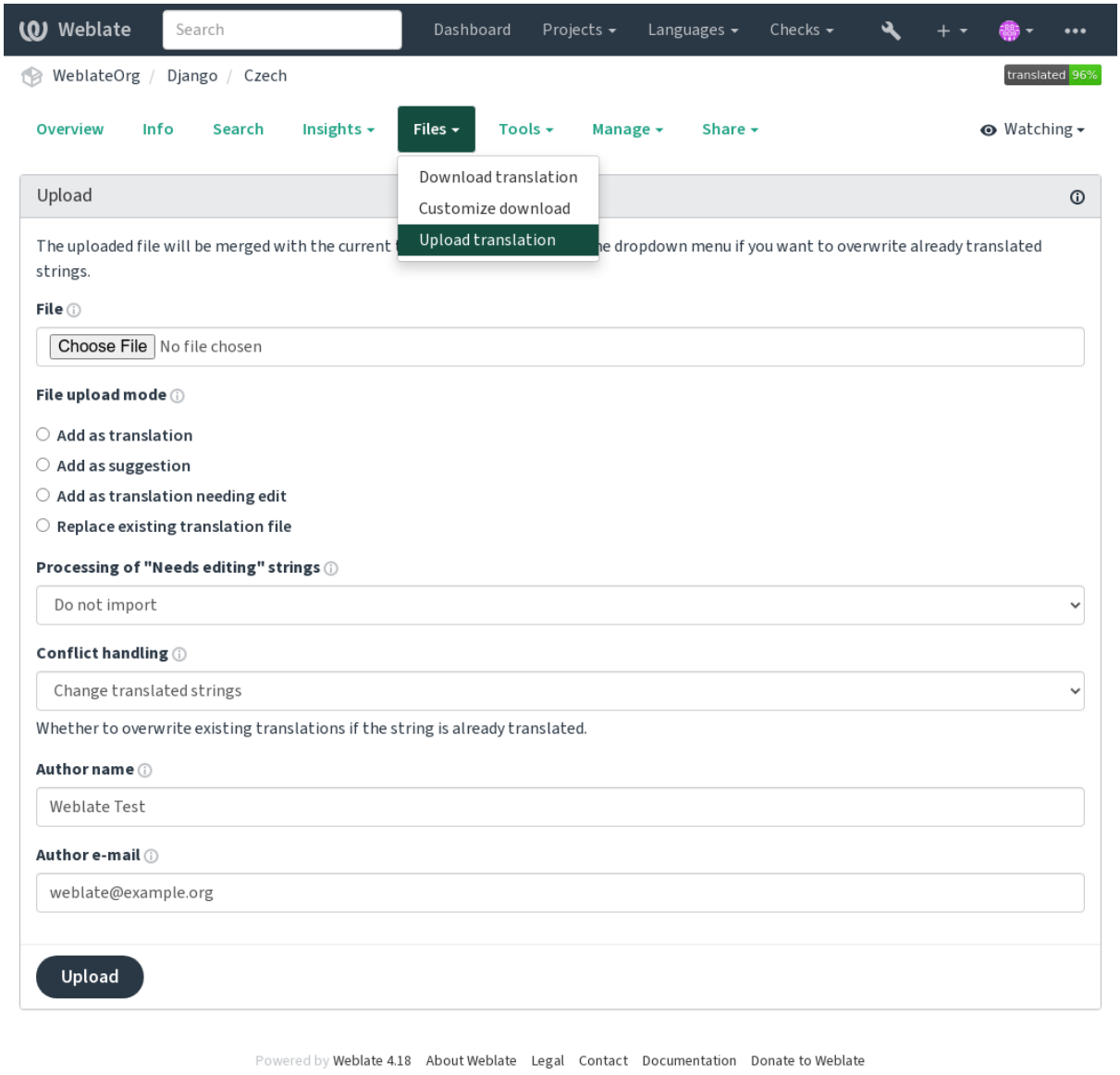


Ver también:

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

1.4.2 Cargar traducciones

Cuando haya efectuado sus cambios, utilice *Cargar traducción* en el menú *Archivos*.



Formatos de archivo admitidos

Es posible cargar archivos en cualquier formato compatible, pero aun así se recomienda utilizar el mismo formato de archivo que el utilizado para la traducción; de lo contrario, algunas funciones podrían no convertirse apropiadamente.

Ver también:

Formatos de archivo admitidos, *Descargar y cargar traducciones*

Métodos de importación

Estas son las elecciones presentadas al cargar archivos de traducción:

Añadir como traducción (`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.

Añadir como sugerencia (`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.

Añadir como traducción que requiere edición (`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.

Reemplazar archivo de traducción existente (`replace`)
: Existing file is replaced with new content. This can lead to loss of existing translations, use with caution.

Actualizar cadenas de origen (`source`)
: Updates source strings in bilingual translation file. This is similar to what *Actualizar archivos PO para que coincidan con POT (msgmerge)* does.

  Solo determinados formatos de archivo admiten esta opción.

Añadir cadenas nuevas (`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*.

  Esta opción está disponible solo cuando se activa *Gestionar cadenas*.

  Only source, translation and key (context) are used from the uploaded file.

Ver también:

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

Conflicts handling

Define el tratamiento que recibirán las cadenas cargadas que ya estén traducidas.

Strings needing edit

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

Overriding authorship

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

1.5 Glosario

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.

Consejo: The glossary terms are not used in quality checks unless you enable that, see *No se ajusta al glosario* for more information.

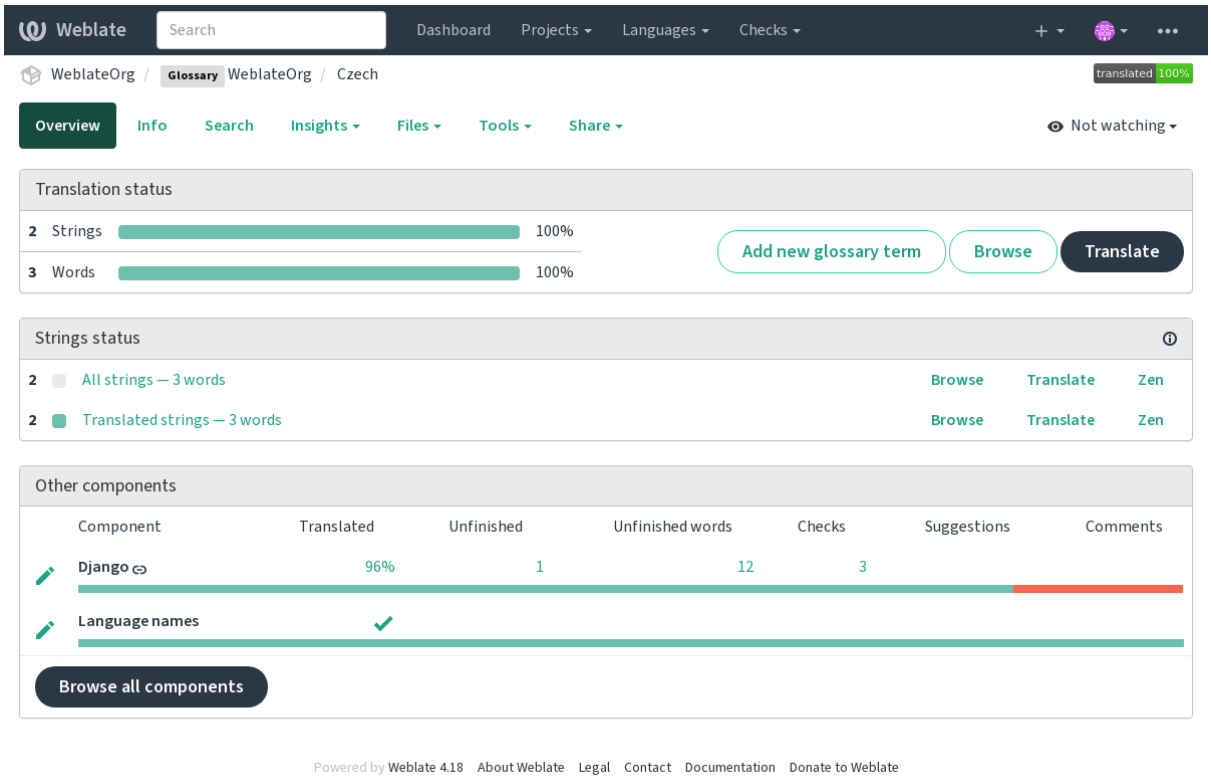
1.5.1 Gestionar glosarios

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

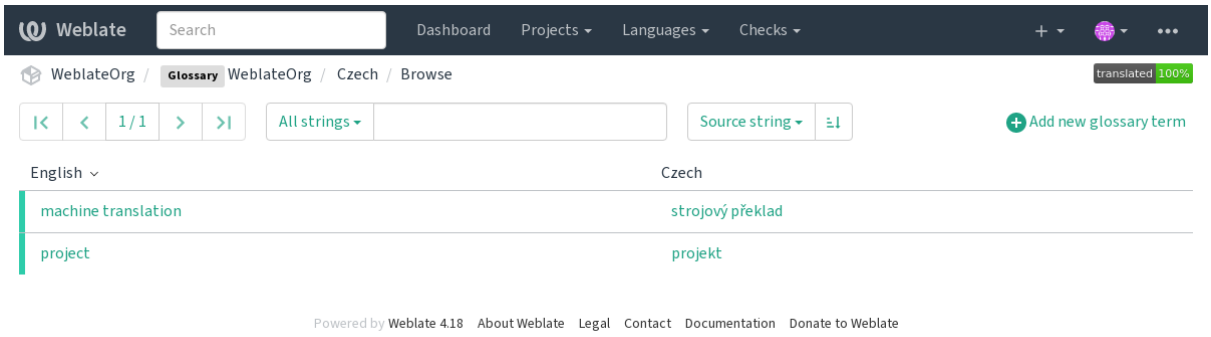
Use any component as a glossary by turning on *Utilizar como glosario*. You can create multiple glossaries for one project.

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 *Compartir en proyectos* from the respective glossary component.

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



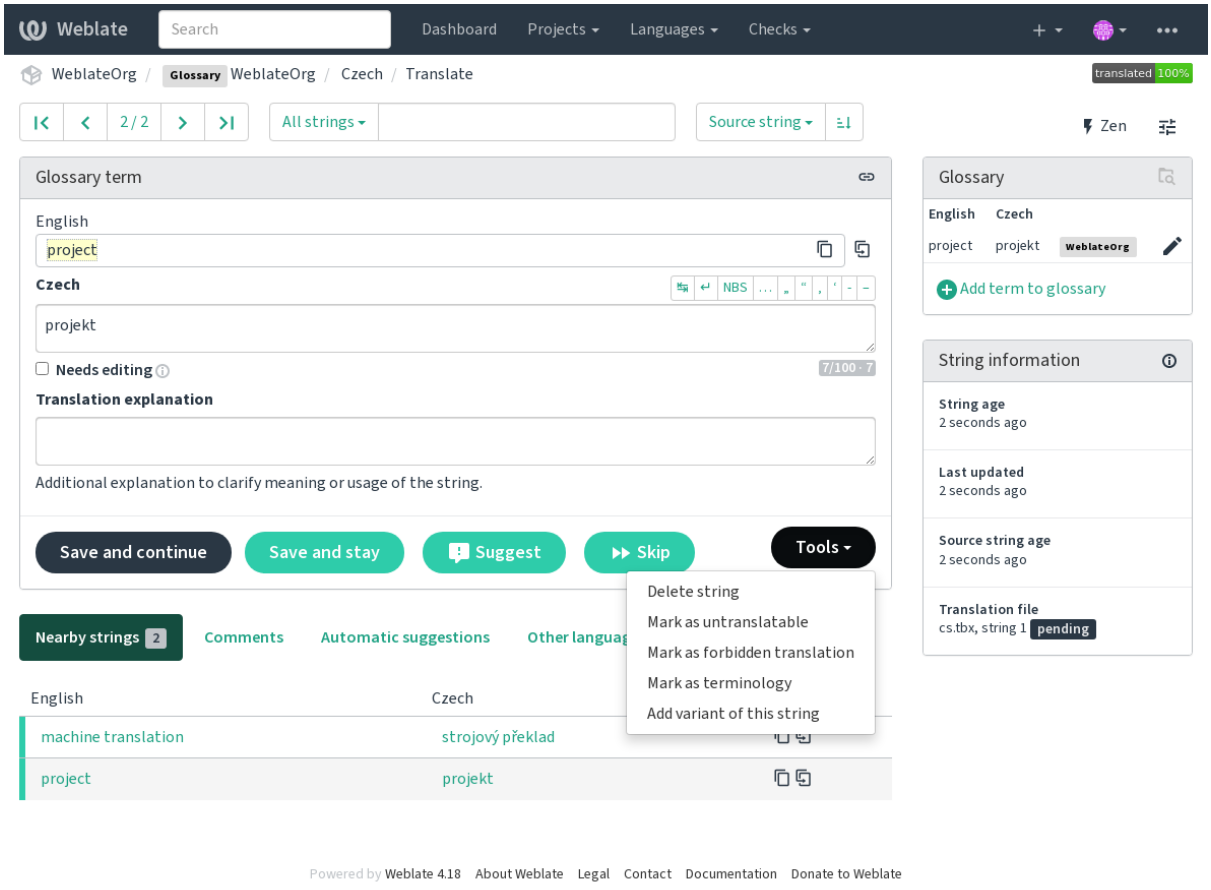
Puede explorar todos los términos del glosario:



or edit them as any translations.

1.5.2 Términos en glosario

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



Untranslatable terms

Nuevo en la versión 4.5.

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.

Ver también:

Personalizar el comportamiento mediante indicadores

Traducciones prohibidas

Nuevo en la versión 4.5.

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.

Ver también:

Personalizar el comportamiento mediante indicadores

Terminología

Nuevo en la versión 4.5.

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.

Ver también:

Personalizar el comportamiento mediante indicadores

Variantes

Las variantes son una forma genérica de agrupar varias cadenas. Todas las variantes de un término aparecerán en el cuadro lateral del glosario mientras traduce.

Consejo: Puede emplearlas para añadir versiones abreviadas de un término.

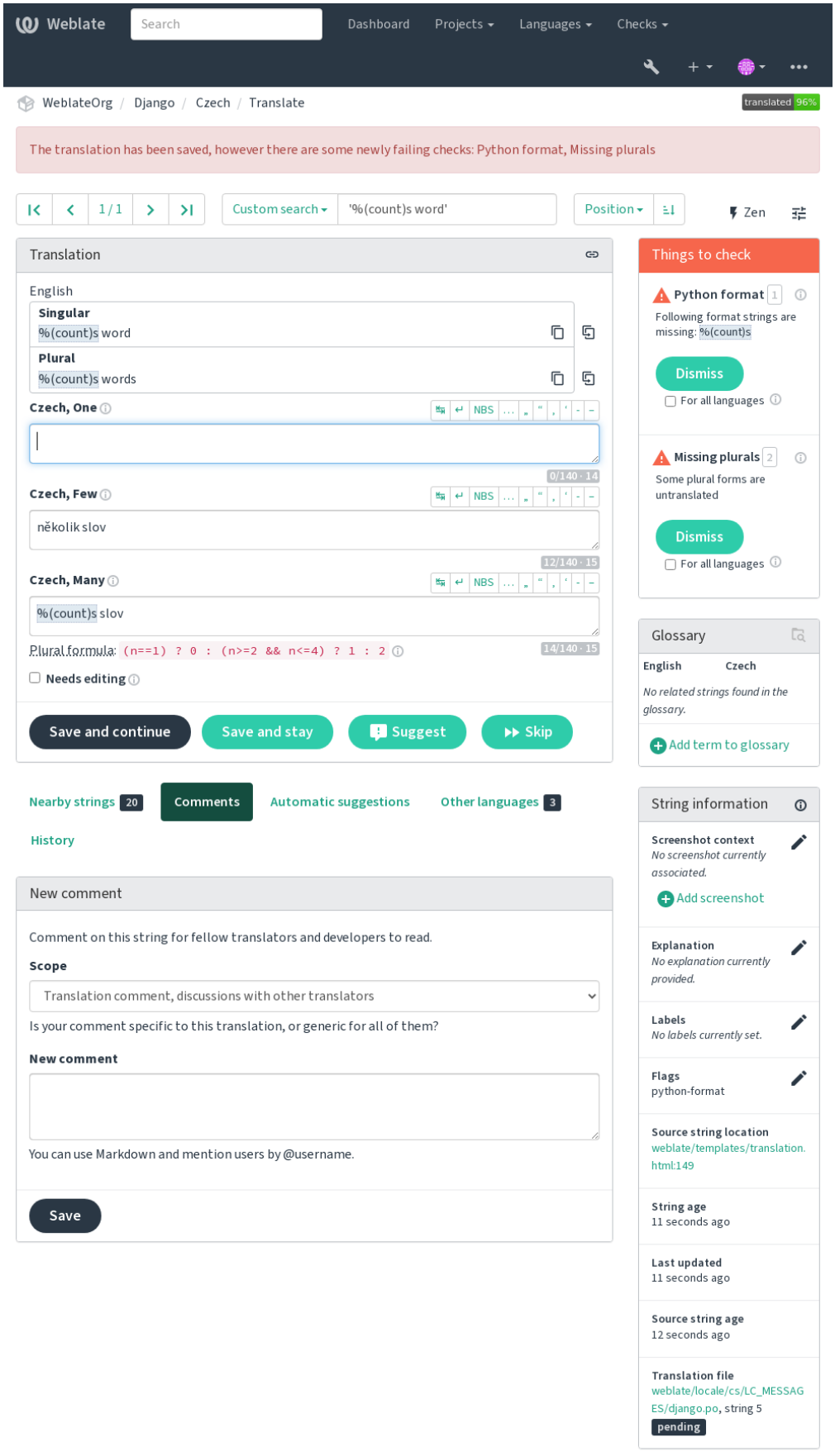
Ver también:

variants

1.6 Comprobaciones y correcciones

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
Search
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Czech / Translate
translated 96%
The translation has been saved, however there are some newly failing checks: Python format, Missing plurals
1 / 1
Custom search
'%(count)s word'
Position
Zen
Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
0/140 · 14
Czech, Few
několik slov
12/140 · 15
Czech, Many
%(count)s slov
14/140 · 15
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
Needs editing
Save and continue
Save and stay
Suggest
Skip
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
Nearby strings 20
Comments
Automatic suggestions
Other languages 3
History
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
11 seconds ago
Last updated
11 seconds ago
Source string age
12 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
pending
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
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.6.1 Correcciones automáticas

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

Ver también:

`AUTOFIX_LIST`

Trailing ellipsis replacer

Replace trailing dots (`...`) with an ellipsis (…) to make it consistent with the source string.

Zero-width space removal

Zero width space is typically not desired in the translation. This fix will remove it unless it is present in the source string as well.

Control characters removal

Removes any control characters from the translation.

Barra devanagari

Replaces wrong full stop in Devanagari by Devanagari danda (।).

Unsafe HTML cleanup

When turned on using a `safe-html` flag it sanitizes HTML markup.

Ver también:

HTML inseguro

Trailing and leading whitespace fixer

Makes leading and trailing whitespace consistent with the source string. The behavior can be fine-tuned using `ignore-begin-space` and `ignore-end-space` flags to skip processing parts of the string.

1.6.2 Comprobaciones de calidad

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.

Ver también:

`CHECK_LIST`, *Personalizar el comportamiento mediante indicadores*

1.6.3 Comprobaciones de traducción

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

Marcación BBCode

Summary
El BBCode en la traducción no coincide con aquél en el original

Alcance
translated strings

Check class
`weblate.checks.markup.BBCodeCheck`

Check identifier
`bbcode`

Flag to ignore
`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.

Nota: El método de detección de BBCode actualmente es bastante sencillo, por lo que es posible que esta comprobación emita falsos positivos.

Palabras consecutivas duplicadas

Nuevo en la versión 4.1.

Summary
El texto contiene la misma palabra dos veces seguidas:

Alcance
translated strings

Check class
`weblate.checks.duplicate.DuplicateCheck`

Check identifier
`duplicate`

Flag to ignore
`ignore-duplicate`

Comprueba que no haya palabras consecutivas duplicadas en una traducción. Esto a menudo señala un error en la traducción.

Consejo: Esta comprobación incluye algunas reglas lingüísticas para evitar falsos positivos. Si encuentra uno, háganoslo saber. Vea *Informar de problemas en Weblate*.

No se ajusta al glosario

Nuevo en la versión 4.5.

Summary
La traducción no sigue los términos definidos en un glosario.

Alcance
translated strings

Check class
`weblate.checks.glossary.GlossaryCheck`

Check identifier
`check_glossary`

Indicador que activar
`check-glossary`

Flag to ignore
`ignore-check-glossary`

Esta comprobación se debe activar mediante el indicador `check-glossary` (vea *Personalizar el comportamiento mediante indicadores*). Considere lo siguiente antes de activarla:

- 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.
- It also utilizes untranslatable glossary terms in *Traducción no modificada*.

Ver también:

Glosario, *Personalizar el comportamiento mediante indicadores*, *Indicadores de traducción*

Espacio duplicado

Summary
La traducción contiene un espacio doble

Alcance
translated strings

Check class
`weblate.checks.chars.DoubleSpaceCheck`

Check identifier
`double_space`

Flag to ignore
`ignore-double-space`

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

La comprobación se anula cuando se encuentran espacios duplicados en la cadena de origen, lo que indica que son intencionales.

Cadenas formateadas

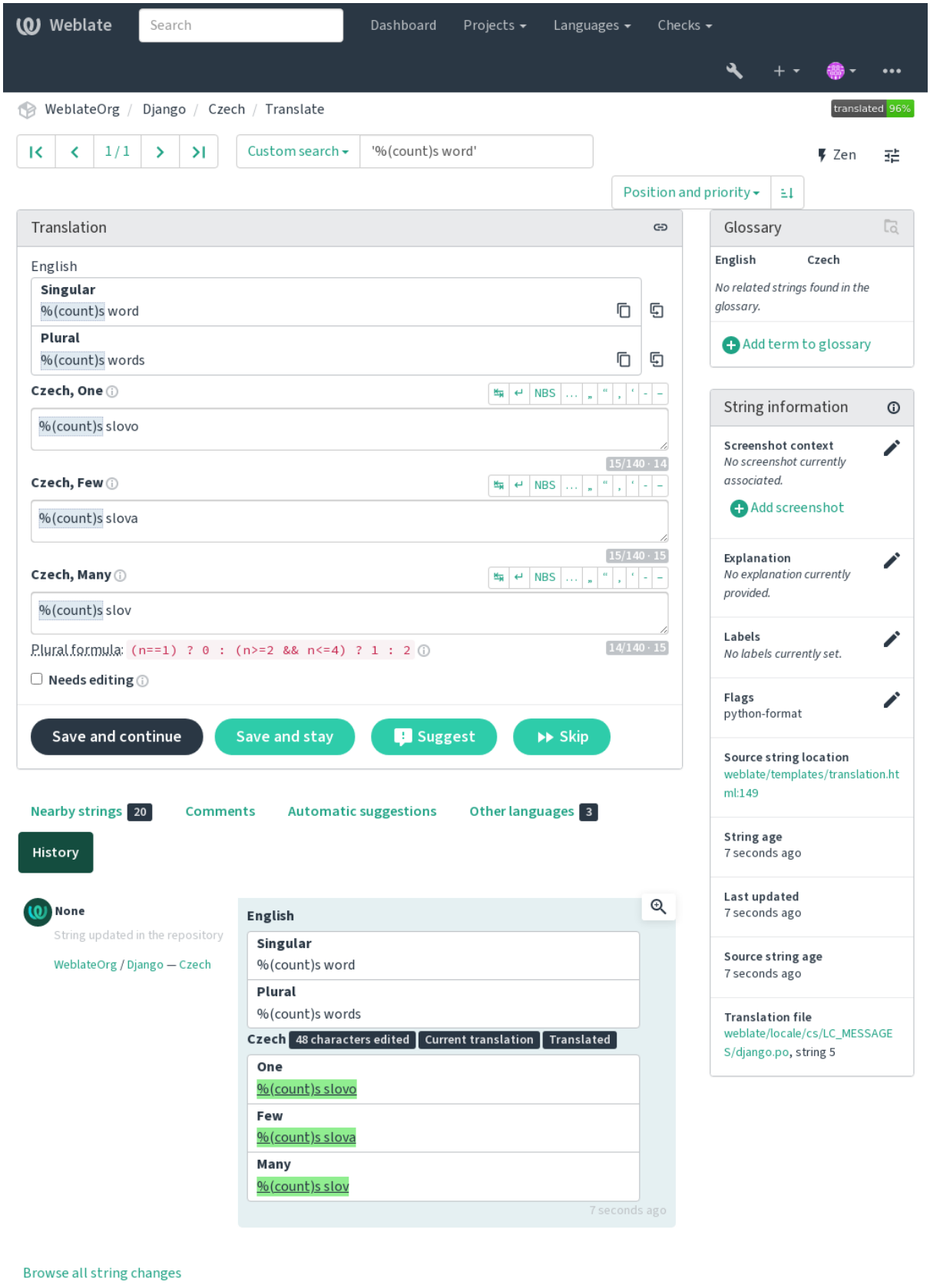
Checks that the formatting in strings is replicated between both source and translation. Omitting format strings in translation usually causes severe problems, so the formatting in strings should usually match the source.

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 string (see *Información adicional sobre las cadenas de origen*) or in a *Configuración de componentes*. Having it defined per component is simpler, but it can lead to false positives in case the string is not interpreted as a formatting string, but format string syntax happens to be used.

Consejo: En caso de que una comprobación específica para un formato no esté disponible en Weblate, puede emplear la genérica *Sustitutivos*.

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


Weblate
Search
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Czech / Translate
translated 96%
1 / 1
Custom search
'%(count)s word'
Zen
Position and priority
Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
%(count)s slovo
15/140 · 14
Czech, Few
%(count)s slova
15/140 · 15
Czech, Many
%(count)s slov
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
14/140 · 15
Needs editing
Save and continue
Save and stay
Suggest
Skip
Nearby strings 20
Comments
Automatic suggestions
Other languages 3
History
None
String updated in the repository
WeblateOrg / Django — Czech
English
Singular
%(count)s word
Plural
%(count)s words
Czech
48 characters edited
Current translation
Translated
One
%(count)s slovo
Few
%(count)s slova
Many
%(count)s slov
7 seconds ago
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
7 seconds ago
Last updated
7 seconds ago
Source string age
7 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
Browse all string changes
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Cadena de interpolación de AngularJS

Summary
La cadena de interpolación de AngularJS no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.angularjs.AngularJSInterpolationCheck`

Check identifier
`angularjs_format`

Indicador que activar
`angularjs-format`

Flag to ignore
`ignore-angularjs-format`

Named format string example
```
Su saldo es de {{amount}} {{ currency }}
```

Ver también:

Cadenas formateadas, AngularJS text interpolation

Formato C

Summary
La cadena en formato C no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.CFormatCheck`

Check identifier
`c_format`

Indicador que activar
`c-format`

Flag to ignore
`ignore-c-format`

Simple format string example
```
Hay %d manzanas
```

Position format string example
```
Su saldo es de %1$d %2$s
```

Ver también:

Cadenas formateadas,
C format strings, C printf format

Formato C#

Summary
La cadena en formato C# no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.CSharpFormatCheck`

Check identifier
`c_sharp_format`

Indicador que activar
`c-sharp-format`

Flag to ignore
`ignore-c-sharp-format`

Position format string example
`Hay {0} manzanas`

Ver también:

Cadenas formateadas, C# String Format

Literales de plantilla ECMAScript

Summary
Los literales de la plantilla ECMAScript no coinciden con los de origen

Alcance
translated strings

Check class
`weblate.checks.format.ESTemplateLiteralsCheck`

Check identifier
`es_format`

Indicador que activar
`es-format`

Flag to ignore
`ignore-es-format`

Interpolation example
`Hay ${number} manzanas`

Ver también:

Cadenas formateadas, Template literals

Interpolación con i18next

Nuevo en la versión 4.0.

Summary
La interpolación de i18next no coincide con aquella en el original

Alcance
translated strings

Check class
`weblate.checks.format.I18NextInterpolationCheck`

Check identifier
`i18next_interpolation`

Indicador que activar
`i18next-interpolation`

Flag to ignore
`ignore-i18next-interpolation`

Interpolation example
`There are {{number}} apples`

Ejemplo de anidamiento
`There are $t(number) apples`

Ver también:

Cadenas formateadas, i18next interpolation

MessageFormat de ICU

Nuevo en la versión 4.9.

Summary
Errores sintácticos o discrepancias en los sustitutivos en las cadenas MessageFormat de ICU.

Alcance
translated strings

Check class
`weblate.checks.icu.ICUMessageFormatCheck`

Check identifier
`icu_message_format`

Indicador que activar
`icu-message-format`

Flag to ignore
`ignore-icu-message-format`

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

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

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

`xml`	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.
`strict-xml`	Enable support for strict XML tags. All < characters must be escaped if they are not part of a tag.
`-highlight`	Desactiva el resalte de los sustitutivos en el editor.
`-require_other`	Disable requiring sub-messages to have an `other` selector.
`-submessage_se`	Skip checking that sub-message selectors match the source.
`-types`	Pasa por alto la comprobación de la coincidencia de los tipos de sustitutivo entre el original y la traducción.
`-extra`	Pasa por alto la comprobación de la no presencia de sustitutivos que no figuren en la cadena de origen.
`-missing`	Pasa por alto la comprobación de la no ausencia de sustitutivos que figuran en la cadena de origen.

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 `<strong>this</strong>`.

Ver también:

Sintaxis MessageFormat de ICU, *Cadenas formateadas*, ICU: Formatting Messages, Format.JS: Message Syntax

Formato Java

Summary
La cadena en formato Java no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.JavaFormatCheck`

Check identifier
`java_printf_format`

Indicador que activar
`java-printf-format`

Flag to ignore
`ignore-java-printf-format`

Simple format string example
`Hay %d manzanas`

Position format string example
`Su saldo es de %1$d %2$s`

Distinto en la versión 4.14: This used to be toggled by `java-format` flag, it was changed for consistency with GNU gettext.

Ver también:

Cadenas formateadas, Java Format Strings

MessageFormat de Java

Summary
La cadena con MessageFormat de Java no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.JavaMessageFormatCheck`

Check identifier
`java_format`

Flag to enable unconditionally
`java-format`

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

Flag to ignore
`ignore-java-format`

Position format string example
`Hay {0} manzanas`

Distinto en la versión 4.14: 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.

Ver también:

Cadenas formateadas, Java MessageFormat

Formato JavaScript

Summary
La cadena en formato JavaScript no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.JavaScriptFormatCheck`

Check identifier
`javascript_format`

Indicador que activar
`javascript-format`

Flag to ignore
`ignore-javascript-format`

Simple format string example
`Hay %d manzanas`

Ver también:

Cadenas formateadas, JavaScript formatting strings

Formato Lua

Summary
La cadena en formato Lua no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.LuaFormatCheck`

Check identifier
`lua_format`

Indicador que activar
`lua-format`

Flag to ignore
`ignore-lua-format`

Simple format string example
`Hay %d manzanas`

Ver también:

Cadenas formateadas, Lua formatting strings

Formato de Object Pascal

Summary
La cadena en formato Object Pascal no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.ObjectPascalFormatCheck`

Check identifier
`object_pascal_format`

Indicador que activar
`object-pascal-format`

Flag to ignore
`ignore-object-pascal-format`

Simple format string example
`Hay %d manzanas`

Ver también:

Cadenas formateadas, Object Pascal formatting strings, Free Pascal formatting strings Delphi formatting strings

Sustitutivos con signo de porcentaje

Nuevo en la versión 4.0.

Summary
Los sustitutivos con signo de porcentaje no coinciden con aquellos en el original

Alcance
translated strings

Check class
`weblate.checks.format.PercentPlaceholdersCheck`

Check identifier
`percent_placeholders`

Indicador que activar
`percent-placeholders`

Flag to ignore
`ignore-percent-placeholders`

Simple format string example
`There are %number% apples`

Ver también:

Cadenas formateadas,

Formato Perl

Summary
La cadena en formato Perl no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.PerlFormatCheck`

Check identifier
`perl_format`

Indicador que activar
`perl-format`

Flag to ignore
`ignore-perl-format`

Simple format string example
`Hay %d manzanas`

Position format string example
`Su saldo es de %1$d %2$s`

Ver también:

Cadenas formateadas, Perl sprintf, Perl Format Strings

Formato PHP

Summary
La cadena en formato PHP no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.PHPFormatCheck`

Check identifier
`php_format`

Indicador que activar
`php-format`

Flag to ignore
`ignore-php-format`

Simple format string example
`Hay %d manzanas`

Position format string example
`Su saldo es de %1$d %2$s`

Ver también:

Cadenas formateadas, PHP sprintf documentation, PHP Format Strings

Formato de llaves de Python

Summary
La cadena en formato de llaves de Python no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.PythonBraceFormatCheck`

Check identifier
`python_brace_format`

Indicador que activar
`python-brace-format`

Flag to ignore
`ignore-python-brace-format`

Simple format string
`There are {} apples`

Named format string example
`Your balance is {amount} {currency}`

Ver también:

Cadenas formateadas, Python brace format, Python Format Strings

Formato Python

Summary
La cadena en formato Python no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.PythonFormatCheck`

Check identifier
`python_format`

Indicador que activar
`python-format`

Flag to ignore
`ignore-python-format`

Simple format string
`Hay %d manzanas`

Named format string example
`Your balance is %(amount)d %(currency)s`

Ver también:

Cadenas formateadas, Python string formatting, Python Format Strings

Formato Qt

Summary
La cadena en formato Qt no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.qt.QtFormatCheck`

Check identifier
`qt_format`

Indicador que activar
`qt-format`

Flag to ignore
`ignore-qt-format`

Position format string example
`There are %1 apples`

Ver también:

Cadenas formateadas, Qt QString::arg()

Formato de plurales de Qt

Summary
La cadena en formato de plurales de Qt no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.qt.QtPluralCheck`

Check identifier
`qt_plural_format`

Indicador que activar
`qt-plural-format`

Flag to ignore
`ignore-qt-plural-format`

Plural format string example
```
There are %Ln apple(s)
```

Ver también:

Cadenas formateadas, Qt i18n guide

Formato Ruby

Summary
La cadena en formato Ruby no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.ruby.RubyFormatCheck`

Check identifier
`ruby_format`

Indicador que activar
`ruby-format`

Flag to ignore
`ignore-ruby-format`

Simple format string example
```
Hay %d manzanas
```

Position format string example
```
Your balance is %1$f %2$s
```

Named format string example
```
Your balance is %+.2<amount>f %<currency>s
```

Named template string
```
Your balance is %{amount} %{currency}
```

Ver también:

Cadenas formateadas, Ruby Kernel#sprintf

Formato Scheme

Summary
La cadena en formato Scheme no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.format.SchemeFormatCheck`

Check identifier
`scheme_format`

Indicador que activar
`scheme-format`

Flag to ignore
`ignore-scheme-format`

Simple format string example
`Hay ~d manzanas`

Ver también:

Cadenas formateadas, Srfi 28, Chicken Scheme format, Guile Scheme formatted output

Formato de Vue I18n

Summary
El formato de Vue I18n no coincide con el de origen

Alcance
translated strings

Check class
`weblate.checks.format.VueFormattingCheck`

Check identifier
`vue_format`

Indicador que activar
`vue-format`

Flag to ignore
`ignore-vue-format`

Named formatting
`There are {count} apples`

Rails i18n formatting
`There are %{count} apples`

Linked locale messages
`@:message.dio @:message.the_world!`

Ver también:

Cadenas formateadas, Vue I18n Formatting, Vue I18n Linked locale messages

Se había traducido

Summary
Esta cadena se ha traducido en el pasado

Alcance
all strings

Check class
`weblate.checks.consistency.TranslatedCheck`

Check identifier
`translated`

Flag to ignore
`ignore-translated`

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

Incoherente

Summary
Esta cadena tiene más de una traducción en este proyecto o no se ha traducido en algunos componentes.

Alcance
all strings

Check class
`weblate.checks.consistency.ConsistencyCheck`

Check identifier
`inconsistent`

Flag to ignore
`ignore-inconsistent`

Weblate comprueba las traducciones de la misma cadena en todas sus ocurrencias dentro de un proyecto para ayudarle a mantener la coherencia.

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 *Permitir propagación de traducciones* turned on.

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

Nota: 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 *Traducción automática* add-on to automate translating of newly added strings which are already translated in another component.

Ver también:

Mantener iguales las traducciones entre los componentes

Kashida utilizado

Nuevo en la versión 3.5.

Summary
Las letras decorativas kashida no deben usarse

Alcance
translated strings

Check class
`weblate.checks.chars.KashidaCheck`

Check identifier
`kashida`

Flag to ignore
`ignore-kashida`

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

Ver también:

Kashida on Wikipedia

Enlaces de Markdown

Nuevo en la versión 3.5.

Summary
Los enlaces de descuento no coinciden con la fuente original

Alcance
translated strings

Check class
`weblate.checks.markup.MarkdownLinkCheck`

Check identifier
`md-link`

Indicador que activar
`md-text`

Flag to ignore
`ignore-md-link`

Markdown links do not match source.

Ver también:

Enlaces de Markdown

Referencias de Markdown

Nuevo en la versión 3.5.

Summary
Las referencias de enlaces de Markdown no coinciden con las de origen

Alcance
translated strings

Check class
`weblate.checks.markup.MarkdownRefLinkCheck`

Check identifier
`md-reflink`

Indicador que activar
`md-text`

Flag to ignore
`ignore-md-reflink`

Markdown link references do not match source.

Ver también:

Markdown links

Sintaxis de Markdown

Nuevo en la versión 3.5.

Summary
La sintaxis de Markdown no coincide con la de origen

Alcance
translated strings

Check class
`weblate.checks.markup.MarkdownSyntaxCheck`

Check identifier
`md-syntax`

Indicador que activar
`md-text`

Flag to ignore
`ignore-md-syntax`

La sintaxis de Markdown no coincide con la de origen

Ver también:

Markdown span elements

Longitud máxima de la traducción

Summary
La traducción no debe exceder la longitud indicada

Alcance
translated strings

Check class
`weblate.checks.chars.MaxLengthCheck`

Check identifier
`max-length`

Indicador que activar
`max-length`

Flag to ignore
`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`.

Consejo: This check looks at number of chars, what might not be the best metric when using proportional fonts to render the text. The *Tamaño máximo de la traducción* 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.

Tamaño máximo de la traducción

Summary
El texto representado de la traducción no debe superar el tamaño indicado

Alcance
translated strings

Check class
`weblate.checks.render.MaxSizeCheck`

Check identifier
`max-size`

Indicador que activar
`max-size`

Flag to ignore
`ignore-max-size`

Nuevo en la versión 3.7.

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

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

You can also configure used font by `font-*` directives (see *Personalizar el comportamiento mediante indicadores*), 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
```

Consejo: You might want to set `font-*` directives in *Configuración de componentes* 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.

Ver también:

Gestionar tipos de letra, *Personalizar el comportamiento mediante indicadores*, *Longitud máxima de la traducción*

\n desiguales

Summary
El número de \n literales en la traducción no coincide con la fuente

Alcance
translated strings

Check class
`weblate.checks.chars.EscapedNewlineCountingCheck`

Check identifier
`escaped_newline`

Flag to ignore
`ignore-escaped-newline`

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

Dos puntos desiguales

Summary
El original o la traducción no acaba con dos puntos

Alcance
translated strings

Check class
`weblate.checks.chars.EndColonCheck`

Check identifier
`end_colon`

Flag to ignore
`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).

Ver también:

Colon on Wikipedia

Puntos suspensivos desiguales

Summary
El original o la traducción no acaba con puntos suspensivos

Alcance
translated strings

Check class
`weblate.checks.chars.EndEllipsisCheck`

Check identifier
`end_ellipsis`

Flag to ignore
`ignore-end-ellipsis`

Comprueba que tanto la cadena de origen como la traducción terminen con puntos suspensivos. Observe que se comprueba solo el carácter real de puntos suspensivos (…), no una secuencia de tres puntos (`...`).

El carácter real de puntos suspensivos a menudo tiene un mejor espaciado al imprimirlo y suena mejor cuando el texto se procesa en un conversor de texto a voz.

Ver también:

Ellipsis on Wikipedia

Signo de exclamación desigual

Summary
El original o la traducción no acaba con un signo de exclamación

Alcance
translated strings

Check class
`weblate.checks.chars.EndExclamationCheck`

Check identifier
`end_exclamation`

Flag to ignore
`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).

Ver también:

Exclamation mark on Wikipedia

Punto final desigual

Summary
El original o la traducción no acaba con un punto

Alcance
translated strings

Check class
`weblate.checks.chars.EndStopCheck`

Check identifier
`end_stop`

Flag to ignore
`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).

Ver también:

Full stop on Wikipedia

Signo de interrogación desigual

Summary
El original o la traducción no acaba con un signo de interrogación

Alcance
translated strings

Check class
`weblate.checks.chars.EndQuestionCheck`

Check identifier
`end_question`

Flag to ignore
`ignore-end-question`

Comprueba que tanto la cadena de origen como la traducción tengan signos de interrogación. La presencia de estos signos también se comprueba en varios idiomas que no los utilizan (armenio, árabe, chino, coreano, japonés, etíope, vai y copto).

Ver también:

Question mark on Wikipedia

Punto y coma desigual

Summary
El original o la traducción no acaba con un punto y coma

Alcance
translated strings

Check class
`weblate.checks.chars.EndSemicolonCheck`

Check identifier
`end_semicolon`

Flag to ignore
`ignore-end-semicolon`

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

Ver también:

Semicolon on Wikipedia

Saltos de renglón desiguales

Summary
La cantidad de saltos de renglón en la traducción no corresponde con la del original

Alcance
translated strings

Check class
`weblate.checks.chars.NewLineCountCheck`

Check identifier
`newline-count`

Flag to ignore
`ignore-newline-count`

Usually newlines are important for formatting program output. Check fails if the number of new lines in translation does not match the source.

Faltan plurales

Summary
Algunos plurales no están traducidos

Alcance
translated strings

Check class
`weblate.checks.consistency.PluralsCheck`

Check identifier
`plurals`

Flag to ignore
`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.

Si no se cumplimentan todas las formas de plural, puede que, en algunos casos, no se muestre nada cuando se utilice la forma de plural.

Sustitutivos

Nuevo en la versión 3.9.

Summary
A la traducción le faltan algunos sustitutivos

Alcance
translated strings

Check class
`weblate.checks.placeholders.PlaceholderCheck`

Check identifier
`placeholders`

Indicador que activar
`placeholders`

Flag to ignore
`ignore-placeholders`

Distinto en la versión 4.3: Puede utilizar expresiones regulares como sustitutivo.

Distinto en la versión 4.13: With the `case-insensitive` flag, the placeholders are not case-sensitive.

A la traducción le faltan algunos sustitutivos. Estos bien se extraen del archivo de traducción, o bien, se definen manualmente mediante el indicador `placeholders`; es posible separarlos mediante dos puntos, y las cadenas con espacios se pueden entrecomillar:

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

Si utiliza alguna sintaxis para sus sustitutivos, podrá utilizar una expresión regular:

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

You can also have case insensitive placeholders:

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

Ver también:

Personalizar el comportamiento mediante indicadores

Espaciado de puntuación

Nuevo en la versión 3.9.

Summary
Falta un espacio indivisible antes del signo de puntuación doble

Alcance
translated strings

Check class
`weblate.checks.chars.PunctuationSpacingCheck`

Check identifier
`punctuation_spacing`

Flag to ignore
`ignore-punctuation-spacing`

Comprueba que haya un espacio indivisible delante de un signo de puntuación de componente doble (a saber: signo de exclamación, signo de interrogación, punto y coma y dos puntos). Esta regla se utiliza solamente en algunos idiomas, como el francés y el bretón, en los que la presencia de este espacio es una norma de microtipografía.

Ver también:

French and English spacing on Wikipedia

Expresión regular

Nuevo en la versión 3.9.

Summary
La traducción no coincide con la expresión regular

Alcance
translated strings

Check class
`weblate.checks.placeholders.RegexCheck`

Check identifier
`regex`

Indicador que activar
`regex`

Flag to ignore
`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$
```

Reused translation

Nuevo en la versión 4.18.

Summary
Different strings are translated the same.

Alcance
translated strings

Check class
`weblate.checks.consistency.ReusedCheck`

Check identifier
`reused`

Flag to ignore
`ignore-reused`

Check that fails if the same translation is used on different source strings. Such translations can be intentional, but can also confuse users.

Mismos plurales

Summary
Algunas formas plurales se traducen de la misma manera

Alcance
translated strings

Check class
`weblate.checks.consistency.SamePluralsCheck`

Check identifier
`same-plurals`

Flag to ignore
`ignore-same-plurals`

Esta comprobación emite un error si se han duplicado algunas formas de plural. En la mayoría de los idiomas deben ser diferentes.

Salto de renglón al inicio

Summary
El original o la traducción no comienza con un salto de renglón

Alcance
translated strings

Check class
`weblate.checks.chars.BeginNewlineCheck`

Check identifier
`begin_newline`

Flag to ignore
`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.

Ver también:

Salto de renglón al final

Espacios iniciales

Summary
El original y la traducción no comienzan con la misma cantidad de espacios

Alcance
translated strings

Check class
`weblate.checks.chars.BeginSpaceCheck`

Check identifier
`begin_space`

Flag to ignore
`ignore-begin-space`

Normalmente, si hay un espacio al inicio de una cadena es para crear sangrías en la interfaz; por esta razón es importante preservarlos.

Salto de renglón al final

Summary
El original o la traducción no acaba con un salto de renglón

Alcance
translated strings

Check class
`weblate.checks.chars.EndNewlineCheck`

Check identifier
`end_newline`

Flag to ignore
`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.

Ver también:

Salto de renglón al inicio

Espacio al final

Summary
El original o la traducción no acaba con un espacio

Alcance
translated strings

Check class
`weblate.checks.chars.EndSpaceCheck`

Check identifier
`end_space`

Flag to ignore
`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.

Traducción no modificada

Summary
El origen y la traducción son idénticos

Alcance
translated strings

Check class
`weblate.checks.same.SameCheck`

Check identifier
`same`

Flag to ignore
`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.

Distinto en la versión 4.17: With `check-glossary` flag (see *No se ajusta al glosario*), the untranslatable glossary terms are excluded from the checking.

Ver también:

No se ajusta al glosario, *Configuración de componentes*, *Personalizar el comportamiento mediante indicadores*

HTML inseguro

Nuevo en la versión 3.9.

Summary
La traducción utiliza marcación HTML insegura

Alcance
translated strings

Check class
`weblate.checks.markup.SafeHTMLCheck`

Check identifier
`safe-html`

Indicador que activar
`safe-html`

Flag to ignore
`ignore-safe-html`

The translation uses unsafe HTML markup. This check has to be enabled using `safe-html` flag (see *Personalizar el comportamiento mediante indicadores*). There is also accompanied autofixer which can automatically sanitize the markup.

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

Ver también:

The HTML check is performed by the Ammonia library.

URL

Nuevo en la versión 3.5.

Summary
La traducción no contiene un URL

Alcance
translated strings

Check class
`weblate.checks.markup.URLCheck`

Check identifier
`url`

Indicador que activar
`url`

Flag to ignore
`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.

Marcación XML

Summary
Las etiquetas XML en la traducción no coinciden con aquellas en el original

Alcance
translated strings

Check class
`weblate.checks.markup.XMLTagsCheck`

Check identifier
`xml-tags`

Flag to ignore
`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.

The check is automatically enabled for XML like strings. You might need to add `xml-text` flag in some cases to force turning it on.

Nota: 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.

Sintaxis XML

Summary
La traducción no constituye XML válido

Alcance
translated strings

Check class
`weblate.checks.markup.XMLValidityCheck`

Check identifier
`xml-invalid`

Flag to ignore
`ignore-xml-invalid`

The XML markup is not valid.

The check is automatically enabled for XML like strings. You might need to add `xml-text` flag in some cases to force turning it on.

Nota: 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.

Espacio de anchura cero

Summary
La traducción contiene espacios de anchura cero adicionales

Alcance
translated strings

Check class
`weblate.checks.chars.ZeroWidthSpaceCheck`

Check identifier
`zero-width-space`

Flag to ignore
`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.

Ver también:

Zero width space on Wikipedia

1.6.4 Source checks

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

Puntos suspensivos

Summary
La cadena utiliza tres puntos (...) en lugar del caracter de puntos suspensivos (…)

Alcance
cadenas de origen

Check class
`weblate.checks.source.EllipsisCheck`

Check identifier
`ellipsis`

Flag to ignore
`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.

Ver también:

Ellipsis on Wikipedia

Sintaxis MessageFormat de ICU

Nuevo en la versión 4.9.

Summary
Errores sintácticos en las cadenas MessageFormat de ICU.

Alcance
cadenas de origen

Check class
`weblate.checks.icu.ICUSourceCheck`

Check identifier
`icu_message_format_syntax`

Indicador que activar
`icu-message-format`

Flag to ignore
`ignore-icu-message-format`

Ver también:

MessageFormat de ICU

Largamente no traducida

Nuevo en la versión 4.1.

Summary
Esta cadena no se tradujo por mucho tiempo

Alcance
cadenas de origen

Check class
`weblate.checks.source.LongUntranslatedCheck`

Check identifier
`long_untranslated`

Flag to ignore
`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.

Varias comprobaciones fallidas

Summary
Las traducciones en varios idiomas tienen comprobaciones fallidas

Alcance
cadenas de origen

Check class
`weblate.checks.source.MultipleFailingCheck`

Check identifier
`multiple_failures`

Flag to ignore
`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.

Varias variables sin nombre

Nuevo en la versión 4.1.

Summary
Existen varias variables sin nombre en la cadena, lo cual impide a los traductores reordenarlas

Alcance
cadenas de origen

Check class
`weblate.checks.format.MultipleUnnamedFormatsCheck`

Check identifier
`unnamed_format`

Flag to ignore
`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.

No pluralizada

Summary
La cadena se utiliza como plural pero no utiliza formas de plural

Alcance
cadenas de origen

Check class
`weblate.checks.source.OptionalPluralCheck`

Check identifier
`optional_plural`

Flag to ignore
`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)
```

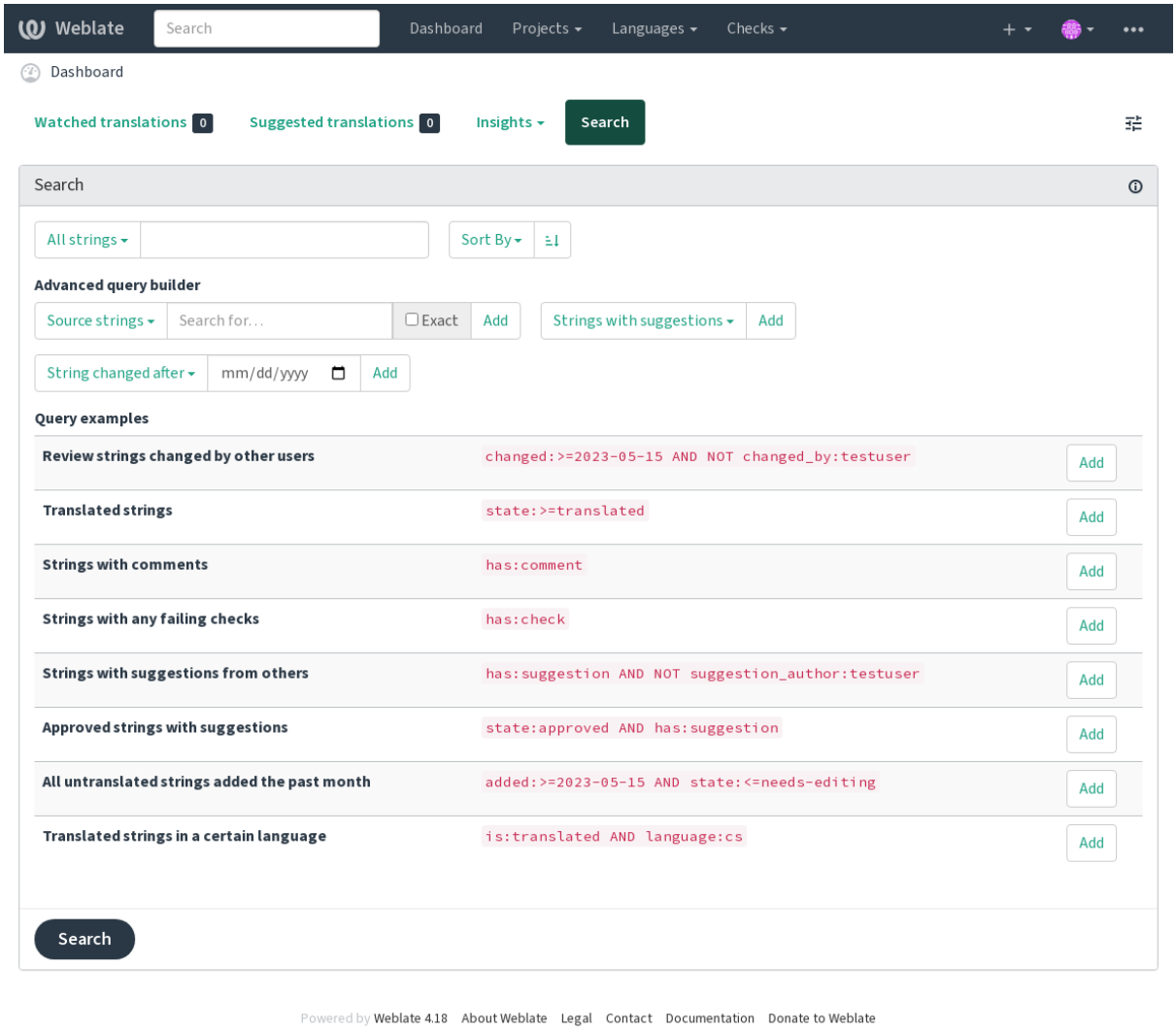
1.7 Búsquedas

1.7.1 Searching for strings

Nuevo en la versión 3.9.

Advanced queries using boolean operations, parentheses, or field specific lookup can be used to find the strings you want.

Cuando no se define ningún campo, la búsqueda se efectúa en las cadenas de origen, de destino y de contexto.



Búsqueda sencilla

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'`.

Campos

`source:TEXTO`
: Búsqueda insensible a mayúsculas y minúsculas de la cadena de origen.

`target:TEXTO`
: Target string case-insensitive search.

`context:TEXTO`
: Context string case-insensitive search.

`key:TEXTO`
: Key string case-insensitive search.

`note:TEXTO`
: Cadena de origen descripción búsqueda insensible a mayúsculas y minúsculas.

location:TEXTO
Location string case-insensitive search.

priority:NÚMERO
String priority.

id:NUMBER
String unique identifier.

position:NUMBER
String position in the translation file.

added:FECHA Y HORA
Timestamp for when the string was added to Weblate.

state:TEXTO
Search for string states (`approved`, `translated`, `needs-editing`, `empty`, `read-only`), supports *Operadores de campo*.

pending:BOOLEANO
String pending for flushing to VCS.

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

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

language:TEXTO
String target language.

component:TEXTO
Component slug or name case-insensitive search, see *«Slug» del componente* and *Nombre de componente*.

project:TEXTO
Project slug, see *«Slug» del URL*.

changed_by:TEXTO
String was changed by author with given username.

changed:FECHA Y HORA
String content was changed on date, supports *Operadores de campo*.

change_time:DATETIME
String was changed on date, supports *Operadores de campo*, unlike `changed` this includes event which don't change content and you can apply custom action filtering using `change_action`.

change_action:TEXT
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 *Searching for changes* for examples.

check:TEXTO
String has failing check, see *Comprobaciones y correcciones* for check identifiers.

dismissed_check:TEXTO
String has dismissed check, see *Comprobaciones y correcciones* for check identifiers.

comment:TEXTO
Search in user comments.

resolved_comment:TEXT
Search in resolved comments.

comment_author:TEXTO
Filter by comment author.

suggestion:TEXTO
Search in suggestions.

suggestion_author:TEXTO
Filter by suggestion author.

explanation:TEXTO
Buscar en las explicaciones.

label:TEXT
Buscar en las etiquetas.

screenshot:TEXT
Buscar en capturas de pantalla.

Operadores booleanos

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

Operadores de campo

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.

position:[10 to 100]
Strings with position between 10 and 100 (inclusive).

Operadores exactos

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`.

Searching for changes

Nuevo en la versión 4.4.

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"`.

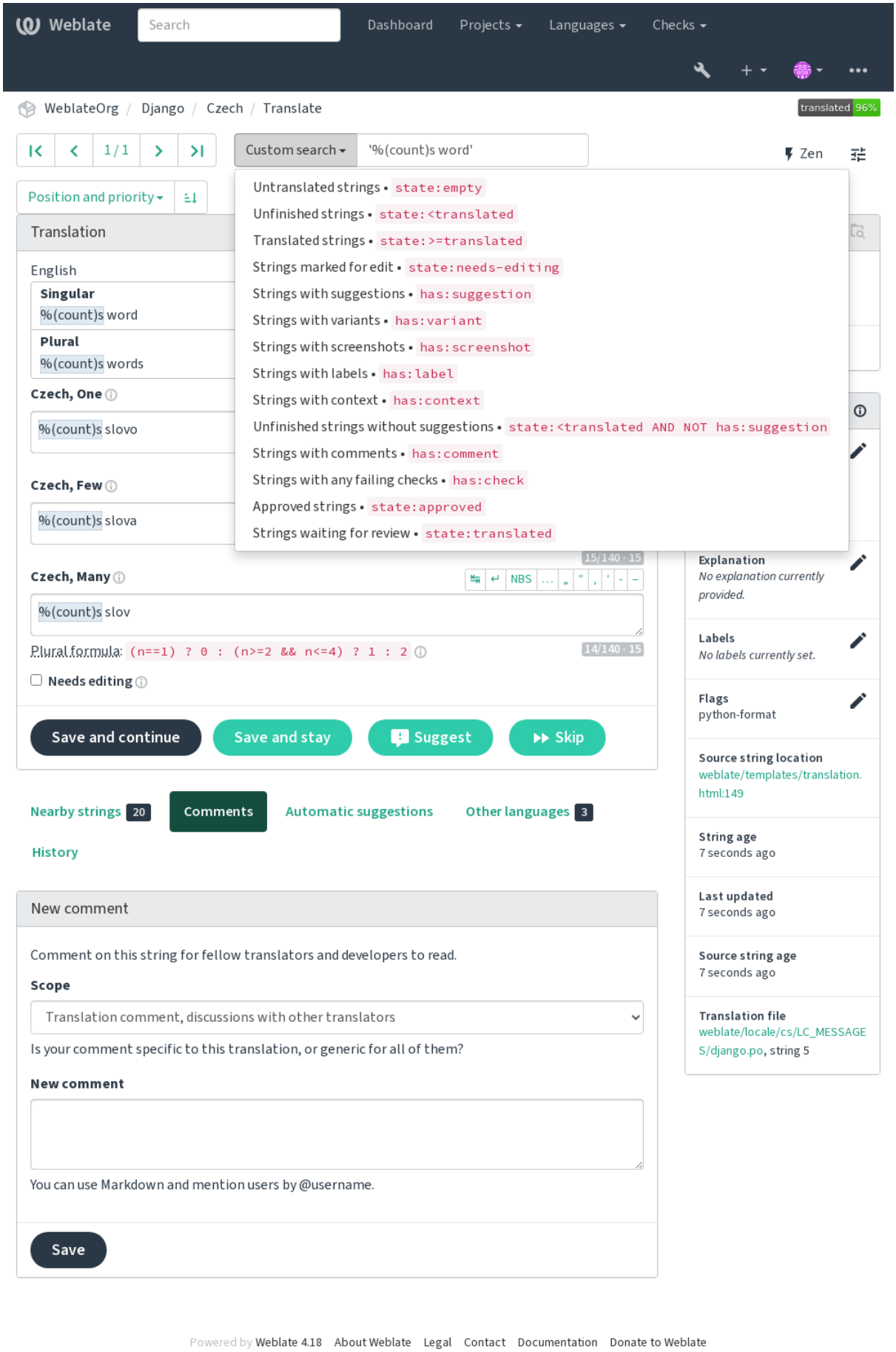
Expresiones regulares

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]"`.

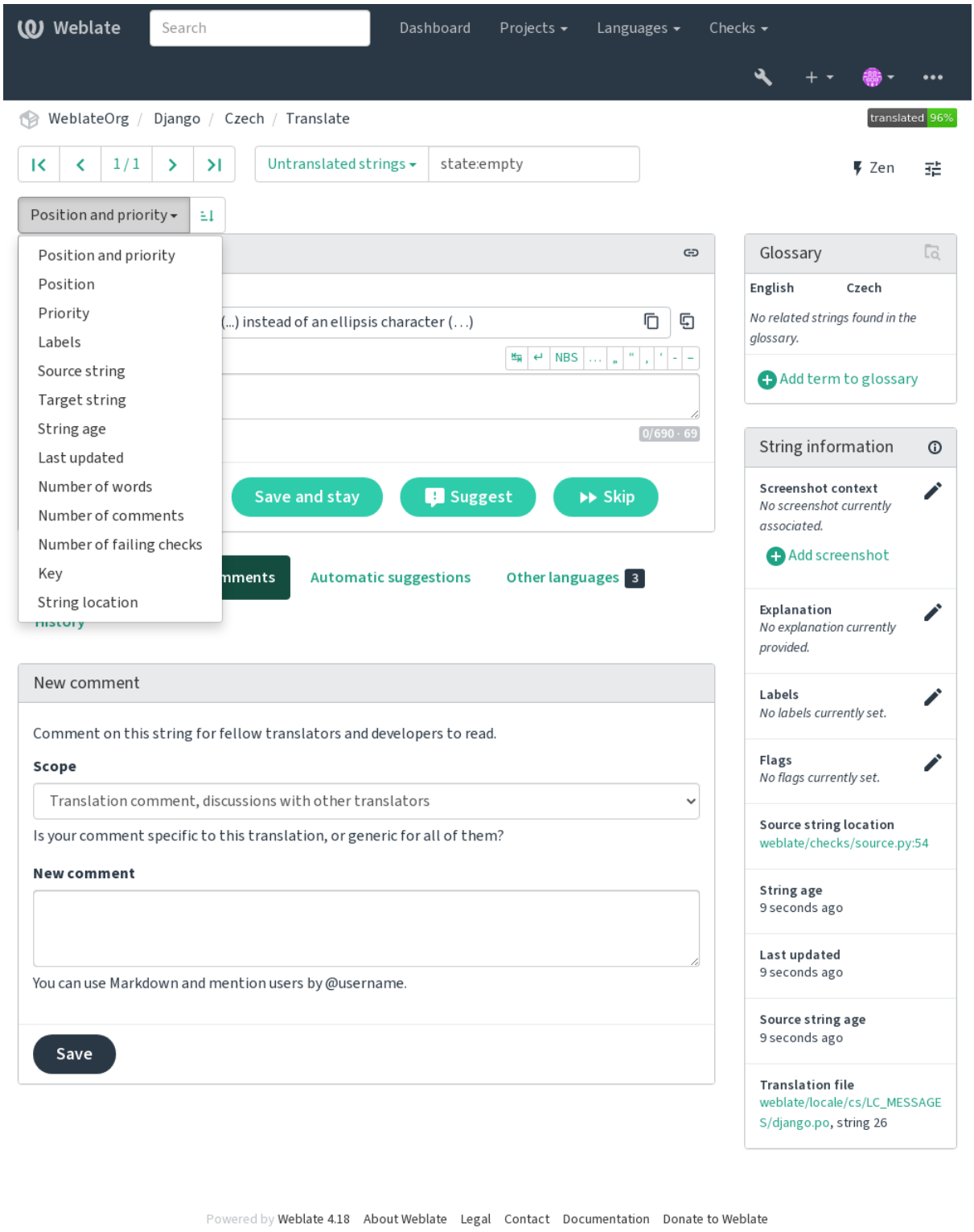
Consultas predefinidas

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


Weblate
Search
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Czech / Translate
translated 96%
1 / 1
Custom search
'%(count)s word'
Zen
Position and priority
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
Strings waiting for review • state:translated
Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
%(count)s slovo
Czech, Few
%(count)s slova
Czech, Many
%(count)s slov
15/140 · 15
14/140 · 15
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
Needs editing
Save and continue
Save and stay
Suggest
Skip
Explanation
No explanation currently provided.
Labels
No labels currently set.
Flags
python-format
Source string location
weblate/templates/translation.html:149
String age
7 seconds ago
Last updated
7 seconds ago
Source string age
7 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
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
Powered by Weblate 4.18 About Weblate Legal Contact Documentation Donate to Weblate

Ordenar los resultados

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



1.7.2 Searching for users

Nuevo en la versión 4.18.

The user browsing has similar search abilities:

username:TEXT
Search in usernames.

full_name:TEXT
Search in full names.

language:TEXTO
User configured translation language (see *Idiomas traducidos*).

joined:DATETIME
String content was changed on date, supports *Operadores de campo*.

translates:TEXT
User has contributed to a given language in the past month.

contributes:TEXT
User has contributed to a given project or component in the past month.

Additional lookups are available in the *Interfaz de gestión*:

is:bot
Search for bots (used for project scoped tokens).

is:active
Search for active users.

email:TEXT
Search by e-mail.

1.8 Flujos de trabajo de traducción

Using Weblate is a process that brings your users closer to you, by bringing you closer to your translators. It is up to you to decide how many of its features you want to make use of.

La siguiente no es una lista completa de maneras de configurar Weblate. Puede basar otros flujos de trabajo en los ejemplos más usuales que se enumeran aquí.

1.8.1 Acceso de traducción

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

In the following chapters, *any user* means a user who has access to the translation. It can be any authenticated user if the project is public, or a user that has a *Translate* permission for the project.

1.8.2 Estados de traducción

Cada cadena traducida se clasifica en uno de los estados siguientes:

No traducidas
Translation is empty, it might or not be stored in the file, depending on the file format.

Necesita edición
Translation needs editing, this is usually the result of a source string change, fuzzy matching or translator action. The translation is stored in the file, depending on the file format it might be marked as needing edit (for example as it gets a `fuzzy` flag in the Gettext file).

Revisión pendiente
La traducción se ha efectuado pero no se ha revisado. Está almacenada en el archivo y es válida.

Aprobadas
Translation has been approved in the review. It can no longer be changed by translators, but only by reviewers. Translators can only add suggestions to it.

This state is only available when reviews are enabled.

Sugerencias
Las sugerencias se almacenan solo en Weblate, no en el archivo de traducción.

Los estados se señalan en los archivos de traducción siempre que es posible.

Consejo: In case file format you use does not support storing states, you might want to use *Marcar traducciones sin cambios como «Necesitan edición»* add-on to flag unchanged strings as needing editing.

Ver también:

Translation types capabilities, *Flujos de trabajo de traducción*

1.8.3 Traducción directa

This is most usual setup for smaller teams, anybody can directly translate. This is also the default setup in Weblate.

- *Cualquier usuario* puede editar las traducciones.
- Suggestions are optional ways to suggest changes, when translators are not sure about the change.

Configuración	Value	Nota
Activar revisiones	desactivada	Configured at project level.
Activar sugerencias	activada	It is useful for users to be able to suggest when they are not sure.
Votar sugerencias	desactivada	
Aceptar sugerencias automáticamente	0	
Grupo de traductores	*Usuario/as*	O bien, *traducir* con *control de acceso por proyecto*.
Grupo de revisores	Sin datos	No se utiliza.

1.8.4 Revisión por pares

Con este flujo de trabajo, cualquiera puede añadir sugerencias, las cuales habrán de recibir aprobación de algún otro miembro antes de que estas se acepten como traducciones.

- *Cualquier usuario* puede añadir sugerencias.
- *Cualquier usuario* puede votar sugerencias.
- Las sugerencias se convierten en traducciones cuando reciben un número de votos determinado previamente.

Configuración	Value	Nota
Activar revisiones	desactivada	Configured at project level.
Activar sugerencias	activada	
Votar sugerencias	desactivada	
Aceptar sugerencias automáticamente	1	Puede establecer un valor más alto para requerir más revisiones por pares.
Grupo de traductores	*Usuario/as*	O bien, *traducir* con *control de acceso por proyecto*.
Grupo de revisores	Sin datos	No se utiliza; todos los traductores revisan.

1.8.5 Revisores dedicados

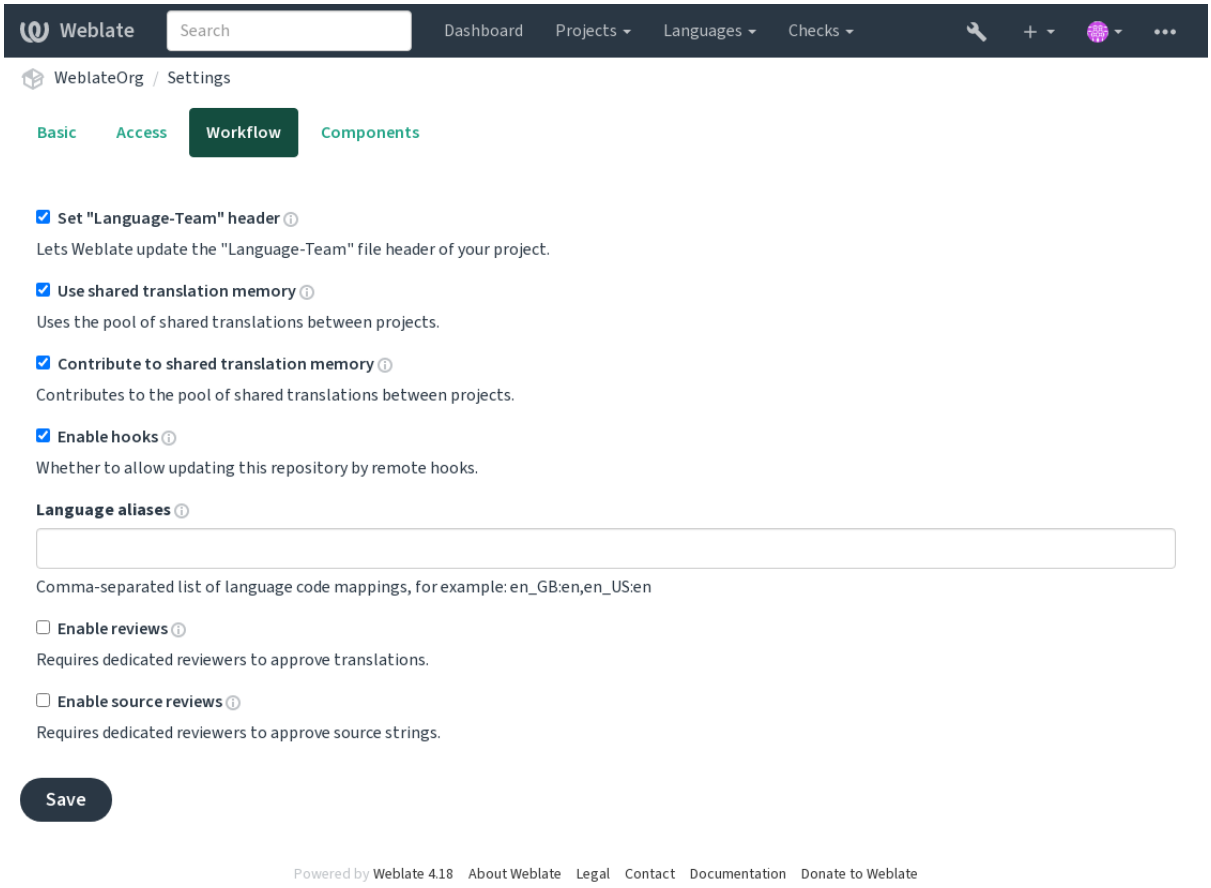
Al activar los revisores dedicados dividirá sus usuarios en dos grupos: uno será capaz de enviar traducciones, y el otro podrá revisarlas para garantizar que sean coherentes y su calidad sea buena.

- *Cualquier usuario* puede editar traducciones no aprobadas.
- Los *revisores* pueden aprobar o desaprobar las cadenas.
- Los *revisores* pueden editar todas las traducciones (incluidas las aprobadas).
- Es posible asimismo emplear las sugerencias para proponer modificaciones a las cadenas aprobadas.

Configuración	Value	Nota
Activar revisiones	activada	Configured at project level.
Activar sugerencias	desactivada	It is useful for users to be able to suggest when they are not sure.
Votar sugerencias	desactivada	
Aceptar sugerencias automáticamente	0	
Grupo de traductores	*Usuario/as*	O bien, *traducir* con *control de acceso por proyecto*.
Grupo de revisores	*Revisores*	Or *Review* with *per-project access control*.

1.8.6 Activar las revisiones

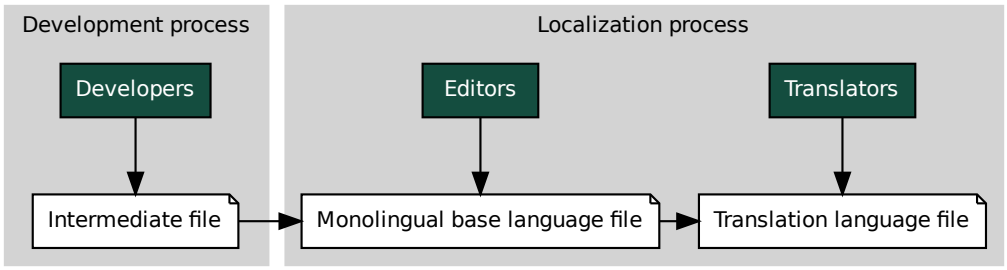
Reviews can be turned on in the project configuration, from the *Workflow* subpage of project settings (to be found in the *Manage* → *Settings* menu):



1.8.7 Quality gateway for the source strings

In many cases the original source language strings are coming from developers, because they write the code and provide initial strings. However developers are often not a native speakers in the source language and do not provide desired quality of the source strings. The intermediate translation can help you in addressing this - there is additional quality gateway for the strings between developers and translators and users.

By setting *Archivo de idioma intermediario*, this file will be used as source for the strings, but it will be edited to source language to polish it. Once the string is ready in the source language, it will be also available for translators to translate into additional languages.



Ver también:

Archivo de idioma intermediario, *Archivo de base monolingüe*, *Formatos bilingües y monolingües*

1.8.8 Source strings reviews

Con *Activar revisiones de origen* habilitado, el proceso de revisión se puede aplicar en las cadenas de origen. Una vez habilitado, los usuarios pueden informar de problemas en las cadenas de origen. El proceso real depende de si utiliza formatos bilingües o monolingües.

Para los formatos monolingües, la revisión de la cadena de origen se comporta de manera similar a *Revisores dedicados*: una vez que se informa de un problema en la cadena de origen, se marca como *Necesita edición*.

Los formatos bilingües no permiten la edición directa de las cadenas de origen (éstas suelen extraerse directamente del código fuente). En este caso, la etiqueta *Source needs review* se adjunta a las cadenas informadas por los traductores. Debe revisar dichas cadenas y editarlas en el código fuente o eliminar la etiqueta.

Ver también:

Formatos bilingües y monolingües, *Revisores dedicados*, labels, *Comentarios*

1.9 Preguntas frecuentes

1.9.1 Configuración

¿Cómo se crea un flujo de trabajo automatizado?

Weblate es capaz de manejar semiautomáticamente todos los procesos de traducción por usted. Si le otorga el acceso de envío a su repositorio, las traducciones pueden integrarse sin interacción por su parte, a menos que ocurra un conflicto de fusión.

1. Set up your Git repository to tell Weblate when there is any change, see *Actuadores de notificación* for info on how to do it.
2. Set a push URL at your *Configuración de componentes* in Weblate, this allows Weblate to push changes to your repository.
3. Turn on *Enviar al consignar* on your *Configuración de componentes* in Weblate, this will make Weblate push changes to your repository whenever they happen at Weblate.

Ver también:

Regionalización continua, *Evitar conflictos de fusión*

¿Cómo se accede a repositorios a través de SSH?

Please see *Accessing repositories* for info on setting up SSH keys.

¿Cómo se arreglan los conflictos de fusión en las traducciones?

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.

Nota: 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
```

Si utiliza una multitud de ramas con Weblate, puede hacer que se efectúen los mismos cambios en todas:

```
# 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
```

(continué en la próxima página)

(proviene de la página anterior)

```
... # 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
```

En el caso de los archivos PO de gettext, existe una forma de fusionar conflictos semiautomáticamente:

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
```

Ver también:

¿Cómo se exporta el repositorio Git que utiliza Weblate?, *Regionalización continua*, *Evitar conflictos de fusión*, *Cliente de Weblate*

¿Cómo traduzco varias ramas al mismo tiempo?

Weblate supports pushing translation changes within one *Configuración de proyectos*. For every *Configuración de componentes* 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.

Una vez que fusione los cambios de Weblate, tal vez tenga que fusionar estas ramas (en función de su flujo de trabajo de desarrollo) y descartar cualquier diferencia:

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

Ver también:

Mantener iguales las traducciones entre los componentes

¿Cómo traducir proyectos multiplataforma?

Weblate admite una amplia gama de formatos de archivo (vea *Formatos de archivo admitidos*) y el camino más sencillo consiste en utilizar el formato nativo para cada plataforma.

Once you have added all platform translation files as components in one project (see *Añadir proyectos y componentes de traducción*), you can utilize the translation propagation feature (turned on by default, and can be turned off in the *Configuración de componentes*) to translate strings for all platforms at once.

Ver también:

Mantener iguales las traducciones entre los componentes

¿Cómo se exporta el repositorio Git que utiliza Weblate?

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.

¿Qué opciones existen para enviar los cambios al proyecto originario?

Esto en gran medida depende de su infraestructura, y Weblate es muy flexible en esta área. A continuación se proporcionan algunos ejemplos de flujos de trabajo en los que se integra Weblate:

- Weblate automatically pushes and merges changes (see *¿Cómo se crea un flujo de trabajo automatizado?*).
- 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.

Por supuesto, puede combinarlos todos como guste.

¿Cómo hago para limitar el acceso de Weblate a las traducciones solamente y no exponer el código fuente?

Es posible utilizar git submodule para separar las traducciones del código fuente sin dejar de tenerlas bajo control de versiones.

1. Cree un repositorio que contenga sus archivos de traducción.
2. Añádalo como submódulo a su base de código:

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

3. Enlace Weblate a este repositorio y ya no necesitará acceder al repositorio que contiene su código fuente.
4. Puede actualizar el repositorio principal con las traducciones provenientes de Weblate así:

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

Consulte la documentación de git submodule para obtener más información.

¿Cómo puedo comprobar que he configurado mi Weblate adecuadamente?

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.

Ver también:

Monitoring Weblate, *Monitoring Celery status*

Why are all commits committed by Weblate <noreply@weblate.org>?

This is the default committer name, configured by *DEFAULT_COMMITER_EMAIL* and *DEFAULT_COMMITER_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.

Ver también:

Configuración de componentes

How to move files in the repository without losing history in Weblate?

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. Lock the affected component in Weblate.
2. Confirme los cambios pendientes y combínelos en el repositorio ascendente.
3. Disable receiving webhooks the *Configuración de proyectos*; 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 *Configuración de componentes* to match the new setup; upon changing configuration, Weblate will fetch the updated repository and notice the changed locations while keeping existing strings.
6. Unlock the component and re-enable hooks in the project configuration.

1.9.2 Uso

How do I review the translations of others?

- There are several review based workflows available in Weblate, see *Flujos de trabajo de traducción*.
- You can subscribe to any changes made in *Notificaciones* 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.

Ver también:

Flujos de trabajo de traducción

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.

Ver también:

report-source, *Comentarios*

¿Cómo puedo utilizar traducciones existentes mientras traduzco?

- Todas las traducciones que se realicen dentro de Weblate pueden reutilizarse gracias a su memoria de traducción compartida.
- Es posible importar archivos de memoria de traducción existentes en Weblate.
- 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.

Ver también:

Configuración de las sugerencias automáticas, *Sugerencias automáticas*, *Memoria de traducción*

Does Weblate update translation files besides translations?

Weblate trata de limitar al mínimo los cambios efectuados en los archivos de traducción. Sin embargo, esto no es posible en ciertos tipos de archivo y se producirán cambios de formato. Si quiere mantener el formato de sus archivos según su predilección, tendrá que servirse de algún actuador preconsigna.

Ver también:

updating-target-files

¿De dónde vienen las definiciones de los idiomas y cómo puedo agregar las mías propias?

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.

Ver también:

Definiciones de idioma

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.

¿Por qué Weblate sigue mostrando cadenas de traducción viejas luego de que he actualizado la plantilla?

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.

Nota: 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.

Ver también:

translations-update, updating-target-files, /devel/gettext, /devel/sphinx

How to handle renaming translation files?

When renaming files in the repository, it can happen that Weblate sees this as removal and adding of the files. This can lead to losing strings history, comments and suggestions.

To avoid that, perform renaming in following steps:

1. Lock the translation component in manage-vcs.
2. Commit pending changes in manage-vcs.
3. Merge Weblate changes to the upstream repository.
4. Disable receiving updates via hooks using *Activar actuadores*.
5. Perform the renaming of the files in the repository.
6. Update the component configuration to match new file names.
7. Enable update hooks and unlock the component.

1.9.3 Solución de problemas

Las peticiones a veces fallan con el error «demasiados archivos abiertos»

Esto sucede en ocasiones cuando su repositorio Git ha crecido en demasía y tiene muchos. Comprimir los repositorios Git mejorará la situación.

La manera más sencilla de hacerlo es ejecutar lo siguiente:

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

Ver también:

DATA_DIR

Cuando entro en el sitio recibo un error «Petición incorrecta (400)»

Previsiblemente, esto se debe a una configuración inadecuada de *ALLOWED_HOSTS*. Aquí deben figurar todos los nombres de anfitrión a los que se desea acceder en su Weblate. Por ejemplo:

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

Ver también:

Allowed hosts setup

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.

- In case the translation to the source language is desired, please change the *Idioma del código fuente* in the component settings. You might want to use *English (Developer)* as a source language, or utilize *Quality gateway for the source strings*.
- En caso de que no se necesite el archivo de traducción correspondiente al idioma de origen, elimínelo del repositorio.
- En caso de que se necesite el archivo de traducción del idioma de origen pero Weblate debería ignorarlo, ajuste el *Filtro de idioma* para excluirlo.

Consejo: 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 *Procesamiento de códigos de idioma* for more details.

1.9.4 Funciones

¿Weblate admite sistemas de control de versiones aparte de Git y Mercurial?

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

You can also use *Auxiliares remotos de Git* in Git to access other VCSes.

Weblate also supports VCS-less operation, see *Archivos locales*.

Nota: 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.

Ver también:

Integración de control de versiones

¿Cómo Weblate da crédito a los traductores?

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.

Ver también:

weblate weblate list_translators, ../devel/reporting

¿Por qué Weblate fuerza la visualización de todos los archivos PO en un único árbol?

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.

Distinto en la versión 4.2: Los traductores pueden traducir todos los componentes de un proyecto en un idioma determinado en su totalidad.

¿Por qué Weblate utiliza códigos de idioma tales como sr_Latn o 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`.

Nota: Weblate defaults to POSIX style language codes with underscore, see *Definiciones de idioma* for more details.

Ver también:

Definiciones de idioma, *Estilo de código de idioma*, adding-translation

1.10 Formatos de archivo admitidos

Weblate supports a wide range of translation formats. Each format is slightly different and provides a different set of capabilities.

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

1.10.1 Recursos de cadenas de Android

Formato de archivo específico de Android para traducir aplicaciones.

Android string resources are monolingual, the *Archivo de base monolingüe* is stored in a different location from the other files – `res/values/strings.xml`.

Nota: 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

Consejo: 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>
```

Ver también:

Android string resources documentation, Android string resources

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`res/values-*/strings.xml`
Archivo de base monolingüe	`res/values/strings.xml`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Recurso de cadenas de Android*

1.10.2 Cadenas de iOS de Apple

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

Apple iOS strings are usually used as monolingual.

Ver también:

Stringsdict format, Apple «strings files» documentation, Message Catalog File Format in PWG 5100.13, Mac OSX strings

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`Resources/*.lproj/Localizable.strings`
Archivo de base monolingüe	`Resources/en.lproj/Localizable.strings` or `Resources/Base.lproj/Localizable.strings`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Cadenas de iOS (UTF-8)*

1.10.3 Archivos de metadatos de tiendas de aplicaciones

Nuevo en la versión 3.5.

Es posible traducir los metadatos utilizados en las catalogaciones de las aplicaciones en varias tiendas. Actualmente, las herramientas siguientes son compatibles:

- Triple-T gradle-play-publisher
- Fastlane
- F-Droid

Los metadatos se componen de varios archivos de texto, los cuales Weblate presentará como cadenas por traducir separadas.

Consejo: In case you don't want to translate certain strings (for example changelogs), mark them read-only (see *Personalizar el comportamiento mediante indicadores*). This can be automated by the *Edición en masa*.

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`fastlane/android/metadata/*`
Archivo de base monolingüe	`fastlane/android/metadata/en-US`
Plantilla para traducciones nuevas	`fastlane/android/metadata/en-US`
Formato de archivo	*Archivos de metadatos de tiendas de aplicaciones*

1.10.4 Archivo ARB

Nuevo en la versión 4.1.

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

Ver también:

JSON, Application Resource Bundle Specification, Internationalizing Flutter apps, updating-target-files, *Personalizar la salida de JSON*, *Limpieza de archivos de traducción*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`lib/l10n/intl_*.arb`
Archivo de base monolingüe	`lib/l10n/intl_en.arb`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Archivo ARB*

1.10.5 Archivos CSV

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 *Archivo de base monolingüe* when your files are monolingual (see *Formatos bilingües y monolingües*).

Consejo: By default, the CSV format does autodetection of file encoding. This can be unreliable in some corner cases and causes performance penalty. Please choose file format variant with encoding to avoid this (for example *CSV file (UTF-8)*).

Advertencia: 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.

Ver también:

CSV

Multivalue CSV file

Nuevo en la versión 4.13.

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

Example files

Archivo de ejemplo:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes* for bilingual CSV	
Máscara de archivos	`locale/*.csv`
Archivo de base monolingüe	*Empty*
Plantilla para traducciones nuevas	`locale/en.csv`
Formato de archivo	*Archivo CSV*

Typical Weblate *Configuración de componentes* for monolingual CSV	
Máscara de archivos	`locale/*.csv`
Archivo de base monolingüe	`locale/en.csv`
Plantilla para traducciones nuevas	`locale/en.csv`
Formato de archivo	*Simple CSV file*

1.10.6 DTD files

Ver también:

Mozilla DTD format

Example DTD file:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`locale/*.dtd`
Archivo de base monolingüe	`locale/en.dtd`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*DTD file*

1.10.7 Open XML de Excel

Nuevo en la versión 3.2.

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

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`path/*.xlsx`
Archivo de base monolingüe	`path/en.xlsx`
Plantilla para traducciones nuevas	`path/en.xlsx`
Formato de archivo	*Excel Open XML*

1.10.8 Flat XML files

Nuevo en la versión 3.9.

Ver también:

Flat XML

Example of a flat XML file:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`locale/*.xml`
Archivo de base monolingüe	`locale/en.xml`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Flat XML file*

1.10.9 Fluent format

Nuevo en la versión 4.8.

Nota: Support for this format is under development. Some features might not yet be available and the behavior can change between releases. Feedback from testing is welcome.

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.

Ver también:

Project Fluent website

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`locales/*/messages.ftl`
Archivo de base monolingüe	`locales/en/messages.ftl`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Fluent file*

1.10.10 gettext de GNU

El formato más ampliamente utilizado para traducir sóftwer libre.

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

Ver también:

../devel/gettext, ../devel/sphinx, Gettext on Wikipedia, PO Files, *Actualizar variable ALL_LINGUAS en el archivo «configure»*, *Personalizar la salida de gettext*, *Actualizar archivo LINGUAS*, *Generar archivos MO*, *Actualizar archivos PO para que coincidan con POT (msgmerge)*

gettext monolingüe

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.

Example files

The bilingual gettext PO file typically looks like this:

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

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

#: weblate/accounts/avatar.py:163
msgctxt "No known user"
msgid "None"
msgstr "Žádný"
```

The monolingual gettext PO file typically looks like this:

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

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

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

While the base language file will be:

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

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

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

Configuración de Weblate

Typical Weblate *Configuración de componentes* for bilingual gettext	
Máscara de archivos	`po/*.po`
Archivo de base monolingüe	*Empty*
Plantilla para traducciones nuevas	`po/messages.pot`
Formato de archivo	*Gettext PO file*

Typical Weblate *Configuración de componentes* for monolingual gettext	
Máscara de archivos	`po/*.po`
Archivo de base monolingüe	`po/en.po`
Plantilla para traducciones nuevas	`po/messages.pot`
Formato de archivo	*Gettext PO file (monolingual)*

1.10.11 Archivos JSON de go-i18n

Nuevo en la versión 4.1.

Distinto en la versión 4.16: Support for v2 variant of this format was added.

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

Nota: Weblate supports the go-i18n JSON v1 and v2 variants. Please choose correct file format matching your environment.

Ver también:

JSON, go-i18n, updating-target-files, *Personalizar la salida de JSON*, *Limpieza de archivos de traducción*,

Configuración de Weblate

Typical Weblate *Configuración de componentes* for v1	
Máscara de archivos	`langs/*.json`
Archivo de base monolingüe	`langs/en.json`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*go-i18n v1 JSON file*

Typical Weblate *Configuración de componentes* for v2	
Máscara de archivos	`langs/*.json`
Archivo de base monolingüe	`langs/en.json`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*go-i18n v2 JSON file*

1.10.12 gotext JSON files

Nuevo en la versión 4.15.1.

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

Ver también:

JSON, I18n in Go: Managing Translations, updating-target-files, *Personalizar la salida de JSON*, *Limpieza de archivos de traducción*,

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`internal/translations/locales/*/messages.gotext.json`
Archivo de base monolingüe	`internal/translations/locales/en-GB/messages.gotext.json`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*gotext JSON file*

1.10.13 Propiedades GWT

Native GWT format for translations.

GWT properties are usually used as monolingual translations.

Ver también:

GWT localization guide, GWT Internationalization Tutorial, Mozilla and Java properties files, updating-target-files, *Formatear el archivo de propiedades de Java*, *Limpieza de archivos de traducción*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`src/app/Bundle_*.properties`
Archivo de base monolingüe	`src/app/Bundle.properties`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*GWT Properties*

1.10.14 Archivos HTML

Nuevo en la versión 4.1.

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

Ver también:

HTML

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`path/*.html`
Archivo de base monolingüe	`path/en.html`
Plantilla para traducciones nuevas	`path/en.html`
Formato de archivo	*HTML file*

1.10.15 i18next JSON files

Distinto en la versión 4.15.1: Support for v4 variant of this format was added.

Consejo: 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.

Nota: 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.

Ver también:

JSON, i18next JSON Format, updating-target-files, *Personalizar la salida de JSON*, *Limpieza de archivos de traducción*

Archivo de ejemplo:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`langs/*.json`
Archivo de base monolingüe	`langs/en.json`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*i18next JSON file v3*

1.10.16 Formato IDML

Nuevo en la versión 4.1.

Nota: Support for this format is under development. Some features might not yet be available and the behavior can change between releases. Feedback from testing is welcome.

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`path/*.idml`
Archivo de base monolingüe	`path/en.idml`
Plantilla para traducciones nuevas	`path/en.idml`
Formato de archivo	*IDML file*

1.10.17 Traducciones en INI

Nuevo en la versión 4.1.

INI file format for translations. Weblate currently supports several variants of JSON translations:

- Monolingual INI files
- *Traducciones para Joomla*
- *Traducciones INI de Inno Setup*

INI translations are usually used as monolingual translations.

Nota: Weblate only extracts keys from sections within an INI file. In case your INI file lacks sections, you might want to use *Traducciones para Joomla* or *Propiedades de Java* instead.

Ver también:

INI Files, *Propiedades de Java*, *Traducciones para Joomla*, *Traducciones INI de Inno Setup*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`language/*.ini`
Archivo de base monolingüe	`language/en.ini`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Archivo INI*

1.10.18 Traducciones INI de Inno Setup

Nuevo en la versión 4.1.

Formato de archivo INI de Inno Setup para las traducciones.

Normalmente, las traducciones de archivos INI de Inno Setup se configuran como monolingües.

Nota: La única diferencia importante con *Traducciones en INI* es que se admiten los sustitutivos `%n` y `%t` para insertar saltos de renglón y tabuladores, respectivamente.

Nota: Por ahora, solo se admiten los archivos en codificación Unicode (`.islu`), no los ANSI (`.isl`).

Ver también:

INI Files, *Traducciones para Joomla*, *Traducciones en INI*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`idioma/*.islu`
Archivo de base monolingüe	`idioma/es.islu`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Archivo INI de Inno Setup*

1.10.19 Propiedades de Java

El formato nativo de Java para las traducciones.

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.

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

Ver también:

Java properties on Wikipedia, Mozilla and Java properties files, *mi18n lang files*, *Propiedades GWT*, updating-target-files, *Formatear el archivo de propiedades de Java*, *Limpieza de archivos de traducción*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`src/app/Bundle_*.properties`
Archivo de base monolingüe	`src/app/Bundle.properties`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Propiedades de Java (ISO-8859-1)*

1.10.20 Traducciones para Joomla

Formato nativo de Joomla para traducciones.

Joomla translations are usually used as monolingual translations.

Ver también:

Mozilla and Java properties files, *Traducciones en INI*, *Traducciones INI de Inno Setup*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`language/*/com_foobar.ini`
Archivo de base monolingüe	`language/en-GB/com_foobar.ini`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Archivo de idioma de Joomla*

1.10.21 Archivos JSON

Distinto en la versión 4.3: The structure of JSON file is properly preserved even for complex situations which were broken in prior releases.

El formato JSON se utiliza más a menudo para traducir aplicaciones creadas en JavaScript.

Weblate currently supports several variants of JSON translations:

- Simple key / value files, used for example by *vue-i18n* or *react-intl*.
- Archivos con claves anidadas.
- *i18next JSON files*
- *Archivos JSON de go-i18n*
- *gotext JSON files*
- *JSON para WebExtension*
- *Archivo ARB*

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

Consejo: 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"
   }
}
```

Ver también:

JSON, updating-target-files, *Personalizar la salida de JSON*, *Limpieza de archivos de traducción*,

Example files

Archivo de ejemplo:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`langs/translation-*.json`
Archivo de base monolingüe	`langs/translation-en.json`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*JSON nested structure file*

1.10.22 Cadenas PHP de Laravel

Distinto en la versión 4.1.

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',
];
```

Ver también:

PHP, Localization in Laravel

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`lang/*/texts.php`
Archivo de base monolingüe	`lang/en/texts.php`
Plantilla para traducciones nuevas	`lang/en/texts.php`
Formato de archivo	*Laravel PHP strings*

1.10.23 mi18n lang files

Nuevo en la versión 4.7.

Formato de archivo que se utiliza para la regionalización de JavaScript en mi18n. Sintácticamente coincide con *Propiedades de Java*.

Ver también:

mi18n, Mozilla and Java properties files, *Propiedades de Java*, updating-target-files, *Formatear el archivo de propiedades de Java*, *Limpieza de archivos de traducción*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`*.lang`
Archivo de base monolingüe	`es-PR.lang`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Archivo .lang de mi18n*

1.10.24 Formato OpenDocument

Nuevo en la versión 4.1.

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

Ver también:

OpenDocument Format

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`path/*.odt`
Archivo de base monolingüe	`path/en.odt`
Plantilla para traducciones nuevas	`path/en.odt`
Formato de archivo	*OpenDocument file*

1.10.25 Cadenas de PHP

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

Weblate currently supports several variants of PHP translations:

- Monolingual PHP strings in various syntax
- *Cadenas PHP de Laravel*

Archivo de ejemplo:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`lang/*/texts.php`
Archivo de base monolingüe	`lang/en/texts.php`
Plantilla para traducciones nuevas	`lang/en/texts.php`
Formato de archivo	*Cadenas de PHP*

1.10.26 .ts de Qt Linguist

Formato de traducción utilizado en aplicaciones programadas con Qt.

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

Ver también:

Qt Linguist manual, Qt .ts, *Formatos bilingües y monolingües*

Configuración de Weblate

Typical Weblate *Configuración de componentes* when using as bilingual	
Máscara de archivos	`i18n/app.*.ts`
Archivo de base monolingüe	*Empty*
Plantilla para traducciones nuevas	`i18n/app.de.ts`
Formato de archivo	*Archivo de traducción de Qt Linguist*

Typical Weblate *Configuración de componentes* when using as monolingual	
Máscara de archivos	`i18n/app.*.ts`
Archivo de base monolingüe	`i18n/app.en.ts`
Plantilla para traducciones nuevas	`i18n/app.en.ts`
Formato de archivo	*Archivo de traducción de Qt Linguist*

1.10.27 ResourceDictionary files

Nuevo en la versión 4.13.

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

Ver también:

Flat XML, *Flat XML files*, updating-target-files, *Limpieza de archivos de traducción*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`Languages/*.xaml`
Archivo de base monolingüe	`Language/en.xaml`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*ResourceDictionary file*

1.10.28 RESX .NET resource files

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.

Ver también:

.NET Resource files (.resx), updating-target-files, *Limpieza de archivos de traducción*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`Resources/Language.*.resx`
Archivo de base monolingüe	`Resources/Language.resx`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Archivo de recursos de .NET*

1.10.29 Ruby YAML files

Ruby i18n YAML files with language as root node.

Ver también:

YAML, *YAML files*

Example Ruby i18n YAML file:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`translations/messages.*.yml`
Archivo de base monolingüe	`translations/messages.en.yml`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Ruby YAML file*

1.10.30 Stringsdict format

Nuevo en la versión 4.8.

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

Ver también:

Cadenas de iOS de Apple, Stringsdict File Format

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`Resources/*.lproj/Localizable.stringsdict`
Archivo de base monolingüe	`Resources/en.lproj/Localizable.stringsdict` or `Resources/Base.lproj/Localizable.stringsdict`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Stringsdict file*

1.10.31 Archivos de subtítulos

Nuevo en la versión 3.7.

Con Weblate puede traducir varios archivos de subtítulos:

- Archivo de subtítulos de SubRip (`*.srt`)
- Archivo de subtítulos de MicroDVD (`*.sub`)
- Archivo de subtítulos Advanced Substation Alpha (`*.ass`)
- Archivo de subtítulos Substation Alpha (`*.ssa`)

Ver también:

Subtitles

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`ruta/*.srt`
Archivo de base monolingüe	`ruta/en.srt`
Plantilla para traducciones nuevas	`ruta/en.srt`
Formato de archivo	*Archivo de subtítulos de SubRip*

1.10.32 TermBase eXchange format

Tabla 1: Supported features

Explicación	Source string explanation is saved and loaded from the `<descrip>` tag, translation string explanation from `<node from="translator">`.

Nuevo en la versión 4.5.

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

Ver también:

TBX on Wikipedia, TBX, *Glosario*

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`tbx/*.tbx`
Archivo de base monolingüe	*Empty*
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*TermBase eXchange file*

1.10.33 Archivos de texto

Nuevo en la versión 4.6.

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

Hay tres variedades de este formato:

- Archivo de texto sin formato
- Archivo de texto DokuWiki
- Archivo de texto MediaWiki

Ver también:

Simple Text Documents

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`path/*.txt`
Archivo de base monolingüe	`path/en.txt`
Plantilla para traducciones nuevas	`path/en.txt`
Formato de archivo	*Plain text file*

1.10.34 JSON para WebExtension

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

Nota: 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.

Ver también:

JSON, Google chrome.i18n, Mozilla Extensions Internationalization

Archivo de ejemplo:

```
{
  "hello": {
    "message": "Ahoj světe!\n",
    "description": "Description",
    "placeholders": {
      "url": {
        "content": "$1",
```

(continué en la próxima página)

(proviene de la página anterior)

```
        "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"
  }
}
```

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`_locales/*/messages.json`
Archivo de base monolingüe	`_locales/en/messages.json`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*Archivo JSON para WebExtension*

1.10.35 Archivos RC de Windows

Distinto en la versión 4.1: Se ha reescrito el código de compatibilidad con los archivos RC de Windows.

Nota: Support for this format is under development. Some features might not yet be available and the behavior can change between releases. Feedback from testing is welcome.

RC files are language files used to localize translatable text, dialogs, menus, for Windows applications.

Ver también:

Windows RC files

Example files

Archivo RC de Windows de ejemplo:

```
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
```

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`lang/*.rc`
Archivo de base monolingüe	`lang/en-US.rc`
Plantilla para traducciones nuevas	`lang/en-US.rc`
Formato de archivo	*Archivo RC*

1.10.36 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 es compatible con XLIFF en varias variantes:

XLIFF translation file
: 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.

XLIFF with gettext extensions
: XLIFF enriched by XLIFF 1.2 Representation Guide for Gettext PO to support plurals.

Ver también:

XML Localization Interchange File Format (XLIFF) specification, XLIFF 1.2 Representation Guide for Gettext PO, XLIFF on Wikipedia, XLIFF,

Estados de traducción

Distinto en la versión 3.3: Weblate ignored the `state` attribute prior to the 3.3 release.

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

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

Durante el guardado, Weblate evita añadir estos atributos salvo que sea necesario:

- 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*.

Ver también:

Revisores dedicados

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.

Por ejemplo:

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

Especificar indicadores de traducción

You can specify additional translation flags (see *Personalizar el comportamiento mediante indicadores*) 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 *Gestionar tipos de letra*).

Ver también:

font attribute in XLIFF 1.2, maxwidth attribute in XLIFF 1.2

Claves de cadena

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.

Configuración de Weblate

Typical Weblate *Configuración de componentes* for bilingual XLIFF	
Máscara de archivos	`localizations/*.xliff`
Archivo de base monolingüe	*Empty*
Plantilla para traducciones nuevas	`localizations/en-US.xliff`
Formato de archivo	*XLIFF Translation File*

Typical Weblate *Configuración de componentes* for monolingual XLIFF	
Máscara de archivos	`localizations/*.xliff`
Archivo de base monolingüe	`localizations/en-US.xliff`
Plantilla para traducciones nuevas	`localizations/en-US.xliff`
Formato de archivo	*XLIFF Translation File*

1.10.37 YAML files

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

Weblate currently supports several variants of YAML translations:

- Archivos con claves anidadas.
- *Ruby YAML files*

Ver también:

YAML, *Ruby YAML files*

Example of a YAML file:

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

Configuración de Weblate

Typical Weblate *Configuración de componentes*	
Máscara de archivos	`translations/messages.*.yml`
Archivo de base monolingüe	`translations/messages.en.yml`
Plantilla para traducciones nuevas	*Empty*
Formato de archivo	*YAML file*

Ver también:

Translation Related File Formats

1.10.38 Detección automática

Weblate tries to detect file format during *Añadir proyectos y componentes de traducción*. The detection might be wrong for different variants of the same serialization format (JSON, YAML, properties) or file encoding, so please verify that *Formato de archivo* is correct before creating the component.

1.10.39 Translation types capabilities

Tabla 2: Capabilities of all supported formats

Formato	Linguality[Página 108, 1]	Plurales[Página 108, 2]	Descriptions[Página 108]	Contexto[Página 108, 4]	Ubicación[Página 108,]	Flags[Página 10]	Estados adicionales[Página 108, 6]
gettext de GNU	bilingual	yes	yes	yes	yes	yes[9]	needs editing
gettext monolingüe	mono	yes	yes	yes	yes	yes[Página 108, 9]	needs editing
XLIFF	both	yes	yes	yes	yes	yes	needs editing, approved
Propiedades de Java	both	no	yes	no	no	no	
mi18n lang files	mono	no	yes	no	no	no	
Propiedades GWT	mono	yes	yes	no	no	no	
Traducciones para Joomla	mono	no	yes	no	yes	no	
.ts de Qt Linguist	both	yes	yes	no	yes	yes	needs editing
Recursos de cadenas de Android	mono	yes	yes[7]	no	no	yes	
Cadenas de iOS de Apple	both	no	yes	no	no	no	
Cadenas de PHP	mono	no[10]	yes	no	no	no	
Archivos JSON	mono	no	no	no	no	no	
i18next JSON files	mono	yes	no	no	no	no	
Archivos JSON de go-i18n	mono	yes	yes	no	no	no	
gotext JSON files	mono	yes	yes	no	yes	no	
Archivo ARB	mono	yes	yes	no	no	no	
JSON para WebExtension	mono	yes	yes	no	no	no	

continué en la próxima página

Tabla 2 – proviene de la página anterior

Formato	Lingua-lity[Página 108, 1]	Plura-les[Página 108, 2]	Descrip-tions[Página 108]	Contex-to[Página 108, 4]	Ubica-ción[Página 108,]	Flags[Página 10]	Estados adiciona-les[Página 108, 6]
RESX .NET resource files	mono	no	yes	no	no	yes	
Resource-Dictionary files	mono	no	no	no	no	yes	
Archivos CSV	both	no	yes	yes	yes	no	needs edi-ting
YAML files	mono	no	no	no	no	no	
Ruby YAML files	mono	yes	no	no	no	no	
DTD files	mono	no	no	no	no	no	
Flat XML files	mono	no	no	no	no	yes	
Archivos RC de Windows	mono	no	yes	no	no	no	
Open XML de Excel	mono	no	yes	yes	yes	no	needs edi-ting
Archivos de meta-datos de tiendas de aplicacio-nes	mono	no	no	no	no	no	
Archivos de subtítulos	mono	no	no	no	yes	no	
Archivos HTML	mono	no	no	no	no	no	
Formato OpenDocu-ment	mono	no	no	no	no	no	
Formato IDML	mono	no	no	no	no	no	
Traduc-ciones en INI	mono	no	no	no	no	no	
Traduccio-nes INI de Inno Setup	mono	no	no	no	no	no	
TermBase eXchange format	bilingual	no	yes	no	no	yes	
Archivos de texto	mono	no	no	no	no	no	
Stringsdict format	mono	yes	no	no	no	no	
Fluent for-mat	mono	no[11]	yes	no	no	no	

Formatos bilingües y monolingües

Both monolingual and bilingual formats are supported. Bilingual formats store two languages in single file—source and translation (typical examples are *gettext de GNU*, *XLIFF* or *Cadenas de iOS de Apple*). 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 *Recursos de cadenas de Android*). 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 *Archivo de base monolingüe* within Weblate, though the naming might vary in your paradigm.

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

String states

Many file formats only differentiate «Untranslated» and «Translated» strings. With some formats it is possible to store more fine-grained state information, such as «Needs editing» or «Approved».

Descripción de cadena de origen

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

Several formats have native support for providing additional info to translators (for example *XLIFF*, *gettext de GNU*, *JSON para WebExtension*, *Archivos CSV*, *Open XML de Excel*, *.ts de Qt Linguist*, *Archivos JSON de go-i18n*, *gotext JSON files*, *Archivo ARB*, *RESX .NET resource files*). Many other formats extract closest comment as source string description.

Explicación

The *Explicación* on strings can be stored and parsed from a few file formats.

Currently supported only in *TermBase eXchange format*.

Ubicación de cadena de origen

La ubicación en código fuente de una cadena puede ayudar a los traductores avezados a entender de qué manera una determinada cadena se utilizará.

This information is typically available in bilingual formats where strings are extracted from the source code using tools. For example *gettext de GNU* and *.ts de Qt Linguist*.

[1] Vea *Formatos bilingües y monolingües*
[2] See *Pluralized strings*
[3] See *Descripción de cadena de origen*
[4] See *Contexto*
[5] See *Ubicación de cadena de origen*
[8] See *Indicadores de traducción*.
[6] See *String states*
[9] The gettext type comments are used as flags.
[7] XML comment placed before the `<string>` element, parsed as a source string description.
[10] The plurals are supported only for Laravel which uses in string syntax to define them, see Localization in Laravel.
[11] Plurals are handled in the syntax of the strings and not exposed as plurals in Weblate.

Indicadores de traducción

Translation flags allow customizing Weblate behavior. Some formats support defining those in the translation file (you can always define them in the Weblate interface, see *Personalizar el comportamiento mediante indicadores*).

This feature is modelled on flags in *gettext de GNU*.

Additionally, for all XML based format, the flags are extracted from the non-standard attribute `weblate-flags`. Additionally `max-length:N` is supported through the `maxwidth` attribute as defined in the XLIFF standard, see *Especificar indicadores de traducción*.

Ver también:

Personalizar el comportamiento mediante indicadores, PO files documentation

Contexto

Context is used to differentiate identical strings in a bilingual format 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).

For monolingual formats the string identifier (often called key) can serve the same purpose and additional context is not necessary.

Pluralized strings

Plurals are necessary to properly localize strings with variable count. The rules depend on a target language and many formats follow CLDR specification for that.

Consejo: Pluralizing strings need proper support from the application framework as well. Choose native format of your platform such as *gettext de GNU*, *Recursos de cadenas de Android* or *Stringsdict format*.

Cadenas de solo lectura

Nuevo en la versión 3.10.

Read-only strings from translation files will be included, but can not be edited in Weblate. This feature is natively supported by few formats (*XLIFF* and *Recursos de cadenas de Android*), but can be emulated in others by adding a `read-only` flag, see *Personalizar el comportamiento mediante indicadores*.

1.10.40 Supporting other formats

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.

Ver también:

Translation Related File Formats

1.11 Integración de control de versiones

Weblate currently supports *Git* (with extended support for *Solicitudes de incorporación de GitHub*, *Solicitudes de fusión de GitLab*, *Solicitudes de incorporación de Gitea*, *Gerrit*, *Desestabilización* and *Bitbucket Server pull requests*) and *Mercurial* as version control back-ends.

1.11.1 Accessing repositories

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.

Accessing repositories from Hosted Weblate

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 *Repositorio de código fuente* and *URL de envío al repositorio* using the SSH protocol (for example `git@github.com:WeblateOrg/weblate.git`).

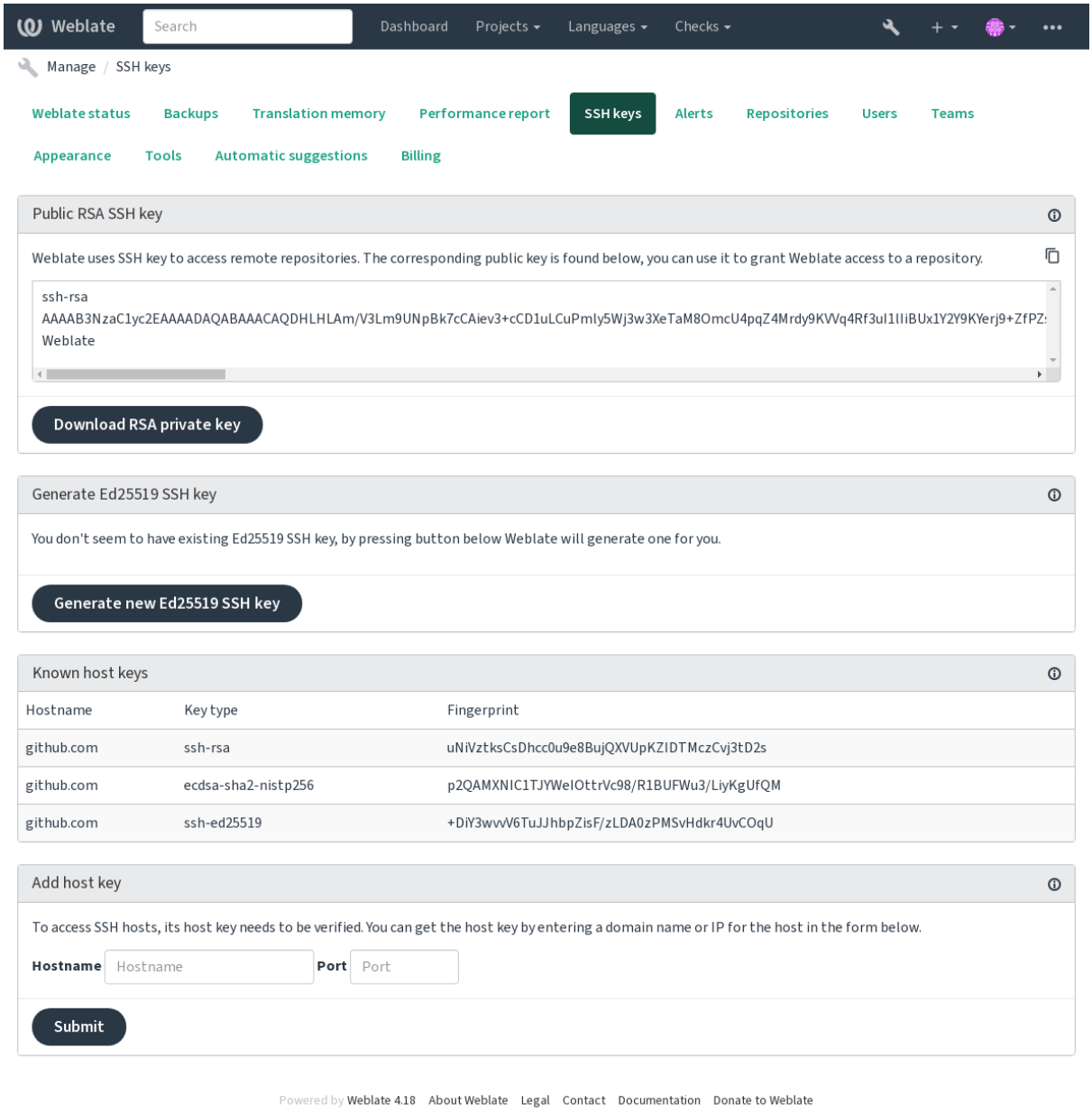
Repositorios SSH

The most frequently used method to access private repositories is based on SSH. Authorize the public Weblate SSH key (see *Clave SSH de Weblate*) to access the upstream repository this way.

Advertencia: On GitHub, each key can only be used once, see *Repositorios en GitHub* and *Accessing repositories from Hosted Weblate*.

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

En caso de que necesite efectuar ajustes, hágalos desde la interfaz administrativa de Weblate:



Clave SSH de Weblate

Distinto en la versión 4.17: Weblate now generates both RSA and Ed25519 SSH keys. Using Ed25519 is recommended for new setups.

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.

Nota: Por ahora, la clave privada SSH correspondiente no puede tener contraseña, así que cerciórese de protegerla adecuadamente.

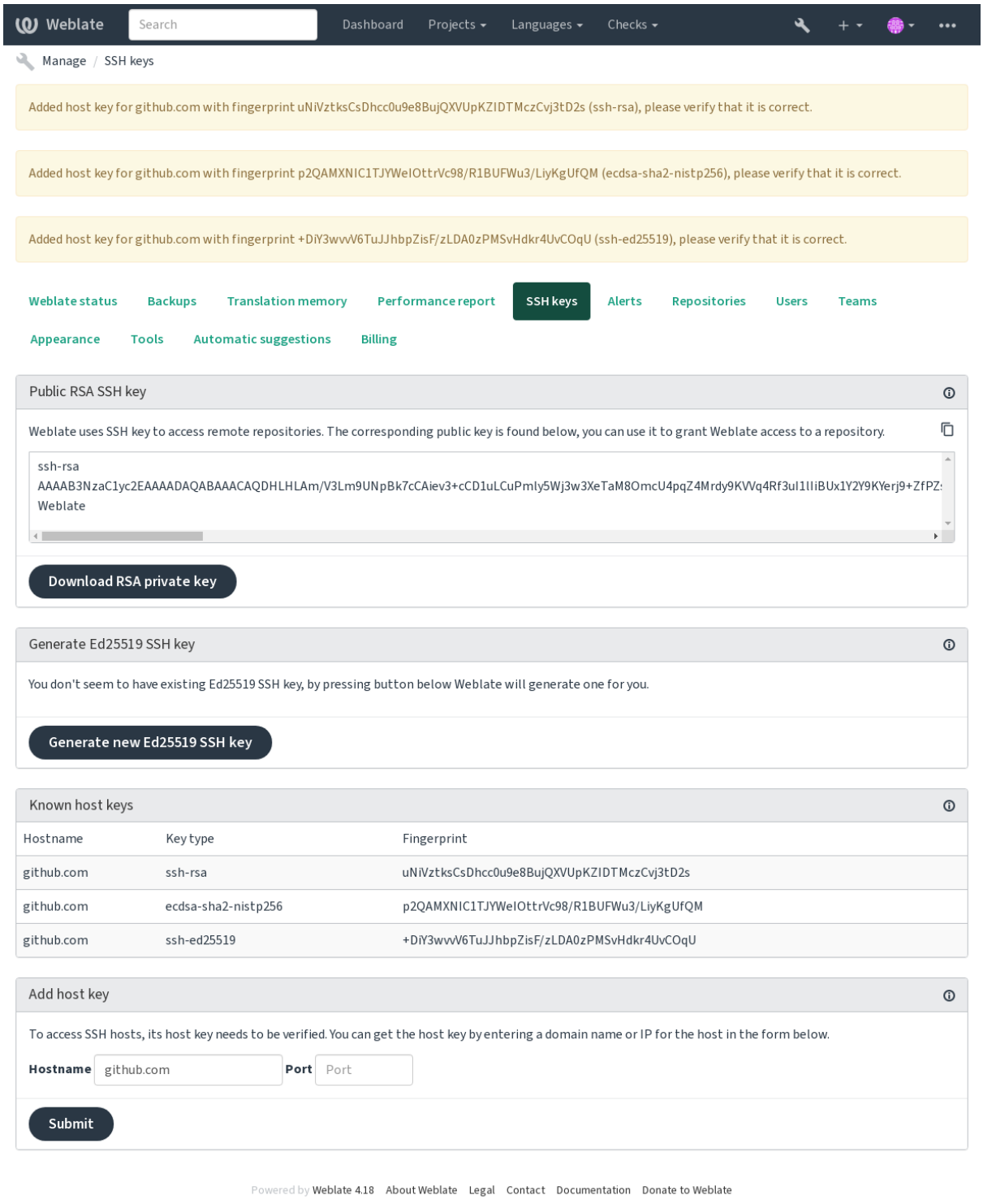
Consejo: Make a backup of the generated private Weblate SSH key.

Verifying SSH host keys

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:



Repositorios en GitHub

Access via SSH is possible (see *Repositorios SSH*), 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 *Rama a la que enviar* 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 *Clave SSH de Weblate*) 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.

Ver también:

Accessing repositories from Hosted Weblate

URL internos de Weblate

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.

Advertencia: 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.

Reasons to use this:

- Ahorra espacio en disco en el servidor, ya que el repositorio se almacena solo una vez.
- Acelera las actualizaciones, ya que se actualiza solo un repositorio.
- 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 *Concentrar consignas de Git*.

Repositorios HTTPS

Para acceder a repositorios HTTPS protegidos, incluya el nombre de usuario y la contraseña en el URL. No se preocupe, Weblate quitará estos datos al mostrar el URL a los usuarios (incluso si se les permite ver el URL del repositorio).

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

Nota: If your username or password contains special characters, those have to be URL encoded, for example `https://user%40example.com:%24password%23@bitbucket.org/`....

Using proxy

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
```

Nota: The proxy configuration needs to be done under user running Weblate (see also *Permisos del sistema de archivos*) and with `HOME=$DATA_DIR/home` (see *DATA_DIR*), otherwise Git executed by Weblate will not use it.

Ver también:

The cURL manpage, Git config documentation

1.11.2 Git

Consejo: Weblate necesita Git 2.12 o posterior.

Ver también:

See *Accessing repositories* for info on how to access different kinds of repositories.

Git con envío forzado

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.

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

Customizing Git configuration

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

Auxiliares remotos de Git

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
```

Advertencia: 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 Solicitudes de incorporación de GitHub

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 *Solicitudes de incorporación de GitHub* 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 *Sistema de control de versiones*.

Ver también:

Enviar cambios efectuados en Weblate, `GITHUB_CREDENTIALS`

1.11.4 Solicitudes de fusión de GitLab

Nuevo en la versión 3.9.

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

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 *Solicitudes de fusión de GitLab* 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 *Sistema de control de versiones*.

Ver también:

Enviar cambios efectuados en Weblate, `GITLAB_CREDENTIALS`

1.11.5 Solicitudes de incorporación de Gitea

Nuevo en la versión 4.12.

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

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 *Solicitudes de incorporación de Gitea* 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 *Sistema de control de versiones*.

Ver también:

Enviar cambios efectuados en Weblate, `GITEA_CREDENTIALS`

1.11.6 Bitbucket Server pull requests

Nuevo en la versión 4.16.

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

Advertencia: This does not support Bitbucket Cloud API.

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

You need to configure API credentials (`BITBUCKETSERVER_CREDENTIALS`) in the Weblate settings to make this work. Once configured, you will see a *Bitbucket Server* option when selecting *Sistema de control de versiones*.

Ver también:

Enviar cambios efectuados en Weblate, `BITBUCKETSERVER_CREDENTIALS`

1.11.7 Solicitudes de fusión de Pagure

Nuevo en la versión 4.3.2.

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

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 *Solicitudes de fusión de Pagure* 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 *Sistema de control de versiones*.

Ver también:

Enviar cambios efectuados en Weblate, `PAGURE_CREDENTIALS`

1.11.8 Gerrit

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.

La documentación de Gerrit tiene los detalles sobre la configuración necesaria para la puesta en marcha de dichos repositorios.

1.11.9 Mercurial

Mercurial es otro sistema de control de versiones que puede utilizar directamente en Weblate.

Nota: Debería funcionar con cualquier versión de Mercurial, pero a veces hay cambios incompatibles en la interfaz de línea de órdenes que quebrantan la integración con Weblate.

Ver también:

See *Accessing repositories* for info on how to access different kinds of repositories.

1.11.10 Desestabilización

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.

Nota: 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.

Datos de acceso de Subversion

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
```

Ver también:

DATA_DIR

1.11.11 Archivos locales

Consejo: Underneath, this uses *Git*. It requires Git installed and allows you to switch to using Git natively with full history of your translations.

Nuevo en la versión 3.8.

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

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

1.12 API REST de Weblate

The API is accessible on the `/api/` URL and it is based on Django REST framework. You can use it directly or by *Cliente de Weblate*.

1.12.1 Autenticación y parámetros genéricos

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.

Parámetros de consulta

- **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).

Encabezados de solicitud

- Accept – the response content type depends on Accept header
- Authorization – optional token to authenticate as `Authorization: Token YOUR-TOKEN`

Encabezados de respuesta

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

Objeto JSON de respuesta

- **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

Códigos de Status

- 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

Fichas de autentificación

Distinto en la versión 4.10: 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.

Authentication examples

Example request:

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

Example response:

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

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

CURL example:

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

Passing Parameters Examples

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

Form request example:

```
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 request example:

```
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 example:

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

CURL JSON example:

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

Limitación de la tasa de API

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.

En el contenedor de Docker, esto se puede configurar usando *WEBLATE_API_RATELIMIT_ANON* y *WEBLATE_API_RATELIMIT_USER*.

The status of rate limiting is reported in following headers:

`X-RateLimit-Limit`	Rate limiting limit of requests to perform
`X-RateLimit-Remaining`	Remaining limit of requests
`X-RateLimit-Reset`	Number of seconds until ratelimit window resets

Distinto en la versión 4.1: Added ratelimiting status headers.

Ver también:

Rate limiting, *Rate limiting*, *WEBLATE_API_RATELIMIT_ANON*, *WEBLATE_API_RATELIMIT_USER*

1.12.2 API Entry Point

`GET /api/`

The API root entry point.

Example request:

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

Example response:

```
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
```

(continué en la próxima página)

(proviene de la página anterior)

```
{
    "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 Usuarios

Nuevo en la versión 4.0.

GET /api/users/

Devuelve un listado de usuarios si tiene permisos para ver y gestionar usuarios. En caso contrario, podrá ver solo sus propios datos.

Ver también:

Users object attributes are documented at *GET /api/users/(str:username)/*.

POST /api/users/

Creates a new user.

Parámetros

- **username** (*string*) – Nombre de usuario
- **full_name** (*string*) – User full name
- **email** (*string*) – User email
- **is_superuser** (*boolean*) – Is user superuser? (optional)
- **is_active** (*boolean*) – Is user active? (optional)
- **is_bot** (*boolean*) – Is user bot? (optional) (used for project scoped tokens)

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

Returns information about users.

Parámetros

- **username** (*string*) – User's username

Objeto JSON de respuesta

- **username** (*string*) – username of a user
- **full_name** (*string*) – full name of a user
- **email** (*string*) – email of a user
- **is_superuser** (*boolean*) – whether the user is a super user
- **is_active** (*boolean*) – whether the user is active
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – date the user is created
- **groups** (*array*) – link to associated groups; see *GET /api/groups/(int:id)/*

Example JSON data:

```
{
    "email": "user@example.com",
    "full_name": "Example User",
    "username": "exampleusername",
```

(continué en la próxima página)

(proviene de la página anterior)


```
    "groups": [
        "http://example.com/api/groups/2/",
        "http://example.com/api/groups/3/"
    ],
    "is_superuser": true,
    "is_active": true,
    "is_bot": false,
    "date_joined": "2020-03-29T18:42:42.617681Z",
    "url": "http://example.com/api/users/exampleusername/",
    "statistics_url": "http://example.com/api/users/exampleusername/statistics/
↪"
}
```


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

Changes the user parameters.

Parámetros

- **username** (*string*) – User's username

Objeto JSON de respuesta

- **username** (*string*) – username of a user
- **full_name** (*string*) – full name of a user
- **email** (*string*) – email of a user
- **is_superuser** (*boolean*) – whether the user is a super user
- **is_active** (*boolean*) – whether the user is active
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – date the user is created

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

Changes the user parameters.

Parámetros

- **username** (*string*) – User's username

Objeto JSON de respuesta

- **username** (*string*) – username of a user
- **full_name** (*string*) – full name of a user
- **email** (*string*) – email of a user
- **is_superuser** (*boolean*) – whether the user is a super user
- **is_active** (*boolean*) – whether the user is active
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – date the user is created

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

Elimina toda la información de la cuenta de usuario y la marca como inactiva.

Parámetros

- **username** (*string*) – User's username

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

Associate groups with a user.

Parámetros

- **username** (*string*) – User's username

Parámetros de Forma

- **string group_id** – The unique group ID

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

Nuevo en la versión 4.13.1.

Eliminar un usuario de un grupo.

Parámetros

- **username** (*string*) – User's username

Parámetros de Forma

- **string group_id** – The unique group ID

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

Enumera estadísticas de una cuenta de usuario.

Parámetros

- **username** (*string*) – User's username

Objeto JSON de respuesta

- **translated** (*int*) – Número de traducciones efectuadas
- **suggested** (*int*) – Número de sugerencias efectuadas
- **uploaded** (*int*) – Número de cargas efectuadas
- **commented** (*int*) – Número de comentarios efectuados
- **languages** (*int*) – Número de idiomas en que puede traducir

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

List subscriptions of a user.

Parámetros

- **username** (*string*) – User's username

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

Associate subscriptions with a user.

Parámetros

- **username** (*string*) – User's username

Objeto JSON de solicitud

- **notification** (*string*) – Nombre de notificación registrada
- **scope** (*int*) – Scope of notification from the available choices
- **frequency** (*int*) – Elecciones de frecuencia de las notificaciones

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

Get a subscription associated with a user.

Parámetros

- **username** (*string*) – User's username
- **subscription_id** (*int*) – Identificador de notificación registrada

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

Edit a subscription associated with a user.

Parámetros

- **username** (*string*) – User's username
- **subscription_id** (*int*) – Identificador de notificación registrada

Objeto JSON de solicitud

- **notification** (*string*) – Nombre de notificación registrada
- **scope** (*int*) – Scope of notification from the available choices
- **frequency** (*int*) – Elecciones de frecuencia de las notificaciones

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

Edit a subscription associated with a user.

Parámetros

- **username** (*string*) – User's username
- **subscription_id** (*int*) – Identificador de notificación registrada

Objeto JSON de solicitud

- **notification** (*string*) – Nombre de notificación registrada
- **scope** (*int*) – Scope of notification from the available choices
- **frequency** (*int*) – Elecciones de frecuencia de las notificaciones

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

Delete a subscription associated with a user.

Parámetros

- **username** (*string*) – User's username
- **subscription_id** – Nombre de notificación registrada
- **subscription_id** – int

1.12.4 Grupos

Nuevo en la versión 4.0.

GET /api/groups/

Devuelve una lista de grupos si tiene permisos para ver y gestionar grupos. En caso contrario, verá solo los grupos a los que pertenece el usuario.

Ver también:

Group object attributes are documented at *GET /api/groups/(int:id)/*.

POST /api/groups/

Creates a new group.

Parámetros

- **name** (*string*) – Nombre de grupo
- **project_selection** (*int*) – Group of project selection from given options
- **language_selection** (*int*) – Group of languages selected from given options
- **defining_project** (*str*) – Enlace al proyecto de la definición, utilizado para *Gestionar el control de acceso por proyecto*; ver *GET /api/projects/(string:project)/*

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

Returns information about group.

Parámetros

- **id** (*int*) – Identificador del grupo

Objeto JSON de respuesta

- **name** (*string*) – nombre de un grupo
- **project_selection** (*int*) – entero que se corresponde a un grupo de proyectos
- **language_selection** (*int*) – entero que se corresponde a un grupo de idiomas
- **roles** (*array*) – link to associated roles; see *GET /api/roles/(int:id)/*
- **projects** (*array*) – link to associated projects; see *GET /api/projects/(string:project)/*
- **components** (*array*) – link to associated components; see *GET /api/components/(string:project)/(string:component)/*
- **componentlists** (*array*) – link to associated componentlist; see *GET /api/component-lists/(str:slug)/*
- **defining_project** (*str*) – Enlace al proyecto de la definición, utilizado para *Gestionar el control de acceso por proyecto*; ver *GET /api/projects/(string:project)/*

Example JSON data:

```
{
    "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*)/

Cambia los parámetros del grupo.

Parámetros

- **id** (*int*) – Identificador del grupo

Objeto JSON de respuesta

- **name** (*string*) – nombre de un grupo
- **project_selection** (*int*) – entero que se corresponde a un grupo de proyectos

- **language_selection** (*int*) – entero que se corresponde a un grupo de idiomas

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

Cambia los parámetros del grupo.

Parámetros

- **id** (*int*) – Identificador del grupo

Objeto JSON de respuesta

- **name** (*string*) – nombre de un grupo
- **project_selection** (*int*) – entero que se corresponde a un grupo de proyectos
- **language_selection** (*int*) – entero que se corresponde a un grupo de idiomas

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

Elimina el grupo.

Parámetros

- **id** (*int*) – Identificador del grupo

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

Associate roles with a group.

Parámetros

- **id** (*int*) – Identificador del grupo

Parámetros de Forma

- **string role_id** – The unique role ID

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

Associate components with a group.

Parámetros

- **id** (*int*) – Identificador del grupo

Parámetros de Forma

- **string component_id** – The unique component ID

DELETE /api/groups/(**int:** *id*)**/components/**
int: *component_id*

Delete component from a group.

Parámetros

- **id** (*int*) – Identificador del grupo
- **component_id** (*int*) – The unique component ID

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

Associate projects with a group.

Parámetros

- **id** (*int*) – Identificador del grupo

Parámetros de Forma

- **string project_id** – The unique project ID

DELETE /api/groups/(**int:** *id*)**/projects/**
int: *project_id*

Delete project from a group.

Parámetros

- **id** (*int*) – Identificador del grupo
- **project_id** (*int*) – The unique project ID

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

Associate languages with a group.

Parámetros

- **id** (*int*) – Identificador del grupo

Parámetros de Forma

- **string language_code** – The unique language code

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

Delete language from a group.

Parámetros

- **id** (*int*) – Identificador del grupo
- **language_code** (*string*) – The unique language code

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

Associate componentlists with a group.

Parámetros

- **id** (*int*) – Identificador del grupo

Parámetros de Forma

- **string component_list_id** – The unique componentlist ID

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

Delete componentlist from a group.

Parámetros

- **id** (*int*) – Identificador del grupo
- **component_list_id** (*int*) – The unique componentlist ID

1.12.5 Roles

GET /api/roles/

Returns a list of all roles associated with user. If user is superuser, then list of all existing roles is returned.

Ver también:

Roles object attributes are documented at *GET /api/roles/(int:id)/*.

POST /api/roles/

Creates a new role.

Parámetros

- **name** (*string*) – Role name
- **permissions** (*array*) – List of codenames of permissions

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

Returns information about a role.

Parámetros

- **id** (*int*) – Role ID

Objeto JSON de respuesta

- **name** (*string*) – Role name
- **permissions** (*array*) – list of codenames of permissions

Example JSON data:

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

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

Changes the role parameters.

Parámetros

- **id** (*int*) – Role's ID

Objeto JSON de respuesta

- **name** (*string*) – Role name
- **permissions** (*array*) – list of codenames of permissions

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

Changes the role parameters.

Parámetros

- **id** (*int*) – Role's ID

Objeto JSON de respuesta

- **name** (*string*) – Role name
- **permissions** (*array*) – list of codenames of permissions

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

Deletes the role.

Parámetros

- **id** (*int*) – Role's ID

1.12.6 Idiomas

GET /api/languages/

Devuelve un listado de todos los idiomas.

Ver también:

Los atributos de objeto de idioma se documentan en *GET /api/languages/(string:language)/*.

POST /api/languages/

Crea un idioma nuevo.

Parámetros

- **code** (*string*) – Nombre del idioma
- **name** (*string*) – Nombre del idioma

- **direction** (*string*) – Dirección del texto
- **population** (*int*) – Número de hablantes
- **plural** (*object*) – Fórmula y número de plurales del idioma

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

Devuelve información relativa a un idioma.

Parámetros

- **language** (*string*) – Código de idioma

Objeto JSON de respuesta

- **code** (*string*) – Código de idioma
- **direction** (*string*) – Dirección del texto
- **plural** (*object*) – Objeto de información de plurales de un idioma
- **aliases** (*array*) – Array of aliases for language

Objeto JSON de solicitud

- **population** (*int*) – Número de hablantes

Example JSON data:

```
{
    "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*)/

Permite cambiar los parámetros del idioma.

Parámetros

- **language** (*string*) – Código del idioma

Objeto JSON de solicitud

- **name** (*string*) – Nombre del idioma
- **direction** (*string*) – Dirección del texto
- **population** (*int*) – Número de hablantes
- **plural** (*object*) – Language plural details

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

Permite cambiar los parámetros del idioma.

Parámetros

- **language** (*string*) – Código del idioma

Objeto JSON de solicitud

- **name** (*string*) – Nombre del idioma
- **direction** (*string*) – Dirección del texto
- **population** (*int*) – Número de hablantes
- **plural** (*object*) – Language plural details

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

Elimina el idioma.

Parámetros

- **language** (*string*) – Código del idioma

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

Devuelve estadísticas sobre un idioma.

Parámetros

- **language** (*string*) – Código de idioma

Ver también:

Returned attributes are described in *Estadísticas*.

1.12.7 Proyectos

GET /api/projects/

Devuelve un listado de todos los proyectos.

Ver también:

Los atributos de objeto de proyecto se documentan en *GET /api/projects/(string:project)/*.

POST /api/projects/

Nuevo en la versión 3.9.

Crea un proyecto nuevo.

Parámetros

- **name** (*string*) – Nombre del proyecto
- **slug** (*string*) – «Slug» del proyecto
- **web** (*string*) – Sitio web del proyecto

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

Devuelve información relativa a un proyecto.

Parámetros

- **project** (*string*) – URL semántico del proyecto

Objeto JSON de respuesta

- **name** (*string*) – nombre del proyecto
- **slug** (*string*) – project slug
- **web** (*string*) – sitio web del proyecto

- **components_list_url** (*string*) – URL a la lista de componentes; vea *GET /api/projects/(string:project)/components/*
- **repository_url** (*string*) – URL al estado del repositorio; vea *GET /api/projects/(string:project)/repository/*
- **changes_list_url** (*string*) – URL a la lista de cambios; vea *GET /api/projects/(string:project)/changes/*
- **translation_review** (*boolean*) – *Activar revisiones*
- **source_review** (*boolean*) – *Activar revisiones de origen*
- **set_language_team** (*boolean*) – *Definir cabecera «Language-Team»*
- **enable_hooks** (*boolean*) – *Activar actuadores*
- **instructions** (*string*) – *Instrucciones de traducción*
- **language_aliases** (*string*) – *Alias de idiomas*

Example JSON data:

```
{
    "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*)/

Nuevo en la versión 4.3.

Edit a project by a PATCH request.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

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

Nuevo en la versión 4.3.

Edit a project by a PUT request.

Parámetros

- **project** (*string*) – URL semántico del proyecto

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

Nuevo en la versión 3.9.

Elimina un proyecto.

Parámetros

- **project** (*string*) – URL semántico del proyecto

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

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

Parámetros

- **project** (*string*) – URL semántico del proyecto

Objeto JSON de respuesta

- **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/*.

Parámetros

- **project** (*string*) – URL semántico del proyecto

Objeto JSON de respuesta

- **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

Example JSON data:

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

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

Efectúa la operación indicada en el repositorio del sistema de control de versiones.

Parámetros

- **project** (*string*) – URL semántico del proyecto

Objeto JSON de solicitud

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

Objeto JSON de respuesta

- **result** (*boolean*) – resultado de la operación

CURL example:

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

JSON request example:

```
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"}
```

Ejemplo de respuesta en JSON

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
```

(continué en la próxima página)

(proviene de la página anterior)

```
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/

Devuelve un listado de componentes de traducción en el proyecto indicado.

Parámetros

- **project** (*string*) – URL semántico del proyecto

Objeto JSON de respuesta

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

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

Nuevo en la versión 3.9.

Distinto en la versión 4.3: The `zipfile` and `docfile` parameters are now accepted for VCS-less components, see *Archivos locales*.

Distinto en la versión 4.6: The cloned repositories are now automatically shared within a project using *URL internos de Weblate*. Use `disable_autoshare` to turn off this.

Crea componentes de traducción en el proyecto indicado.

Consejo: Use *URL internos de Weblate* when creating multiple components from a single VCS repository.

Nota: Most of the component creation happens in the background. Check the `task_url` attribute of created component and follow the progress there.

Parámetros

- **project** (*string*) – URL semántico del proyecto

Parámetros de Forma

- **file zipfile** – ZIP file to upload into Weblate for translations initialization
- **file docfile** – Documento para traducir
- **boolean disable_autoshare** – Disables automatic repository sharing via *URL internos de Weblate*.

Objeto JSON de solicitud

- **object** – Parámetros de componentes, consulte *GET /api/components/(string:project)/(string:component)/*

Objeto JSON de respuesta

- **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*.

Ejemplo de la solicitud del formulario CURL:

```
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/
```

Ejemplo de solicitud CURL JSON:

```
curl \
    --data-binary '{
        "branch": "main",
        "file_format": "po",
        "filemask": "po/*.po",
        "name": "Weblate",
        "slug": "weblate",
        "repo": "https://github.com/WeblateOrg/hello.git",
        "template": "",
        "new_base": "po/hello.pot",
        "vcs": "git"
    }' \
    -H "Content-Type: application/json" \
    -H "Authorization: Token TOKEN" \
    http://example.com/api/projects/hello/components/
```

JSON request to create a new component from Git:

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

{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "https://github.com/WeblateOrg/hello.git",
    "template": "",
    "new_base": "po/hello.pot",
    "vcs": "git"
}
```

JSON request to create a new component from another one:

```
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",
```

(continué en la próxima página)

(proviene de la página anterior)

```
    "repo": "weblate://weblate/hello",
    "template": "",
    "new_base": "po/hello.pot",
    "vcs": "git"
}
```

Ejemplo de respuesta en JSON

```
HTTP/1.0 200 OK
Date: Tue, 12 Apr 2016 09:32:50 GMT
Server: WSGIServer/0.1 Python/2.7.11+
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "slug": "weblate",
    "project": {
        "name": "Hello",
        "slug": "hello",
        "source_language": {
            "code": "en",
            "direction": "ltr",
             "population": 159034349015,
            "name": "English",
            "url": "http://example.com/api/languages/en/",
            "web_url": "http://example.com/languages/en/"
        },
        "url": "http://example.com/api/projects/hello/",
        "web": "https://weblate.org/",
        "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
}
```

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

Returns paginated statistics for all languages within a project.

Nuevo en la versión 3.8.

Parámetros

- **project** (*string*) – URL semántico del proyecto

Objeto JSON de respuesta

- **results** (*array*) – array of translation statistics objects
- **language** (*string*) – nombre del idioma

- **code** (*string*) – código del idioma
- **total** (*int*) – número total de cadenas
- **translated** (*int*) – número de cadenas traducidas
- **translated_percent** (*float*) – porcentaje de cadenas traducidas
- **total_words** (*int*) – número total de palabras
- **translated_words** (*int*) – número de palabras traducidas
- **words_percent** (*float*) – porcentaje de palabras traducidas

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

Devuelve estadísticas sobre un proyecto.

Nuevo en la versión 3.8.

Parámetros

- **project** (*string*) – URL semántico del proyecto

Ver también:

Returned attributes are described in *Estadísticas*.

1.12.8 Componentes

Consejo: Utilice *POST /api/projects/(string:project)/components/* para crear nuevos componentes.

GET /api/components/

Devuelve un listado de componentes de traducción.

Ver también:

Component object attributes are documented at *GET /api/components/(string:project)/(string:component)/*.

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

Devuelve información relativa al componente de traducción.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de respuesta

- **project** (*object*) – el proyecto de traducción; vea *GET /api/projects/(string:project)/*
- **name** (*string*) – *Nombre de componente*
- **slug** (*string*) – *«Slug» del componente*
- **vcs** (*string*) – *Sistema de control de versiones*
- **repo** (*string*) – *Repositorio de código fuente*
- **git_export** (*string*) – *URL de repositorio exportado*
- **branch** (*string*) – *Rama del repositorio*
- **push_branch** (*string*) – *Rama a la que enviar*

- **filemask** (*string*) – *Máscara de archivos*
- **template** (*string*) – *Archivo de base monolingüe*
- **edit_template** (*string*) – *Editar archivo de base*
- **intermediate** (*string*) – *Archivo de idioma intermediario*
- **new_base** (*string*) – *Plantilla para traducciones nuevas*
- **file_format** (*string*) – *Formato de archivo*
- **license** (*string*) – *Licencia de la traducción*
- **agreement** (*string*) – *Acuerdo de contribuidor*
- **new_lang** (*string*) – *Adición de traducciones nuevas*
- **language_code_style** (*string*) – *Estilo de código de idioma*
- **source_language** (*object*) – objeto de idioma de origen; vea `GET /api/languages/(string:language)/`
- **push** (*string*) – *URL de envío al repositorio*
- **check_flags** (*string*) – *Indicadores de traducción*
- **priority** (*string*) – *Prioridad*
- **enforced_checks** (*string*) – *Comprobaciones obligatorias*
- **restricted** (*string*) – *Acceso restringido*
- **repoweb** (*string*) – *Explorador del repositorio*
- **report_source_bugs** (*string*) – *Dirección para informar de errores en las cadenas de origen*
- **merge_style** (*string*) – *Estilo de fusión*
- **commit_message** (*string*) – *Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud*
- **add_message** (*string*) – *Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud*
- **delete_message** (*string*) – *Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud*
- **merge_message** (*string*) – *Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud*
- **addon_message** (*string*) – *Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud*
- **pull_message** (*string*) – *Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud*
- **allow_translation_propagation** (*string*) – *Permitir propagación de traducciones*
- **enable_suggestions** (*string*) – *Activar sugerencias*
- **suggestion_voting** (*string*) – *Votar sugerencias*
- **suggestion_autoaccept** (*string*) – *Aceptar sugerencias automáticamente*
- **push_on_commit** (*string*) – *Enviar al consignar*
- **commit_pending_age** (*string*) – *Antigüedad de cambios por consignar*
- **auto_lock_error** (*string*) – *Bloquear al producirse un error*
- **language_regex** (*string*) – *Filtro de idioma*

- **variant_regex** (*string*) – *Expresión regular de variantes*
- **is_glossary** (*bool*) – *Utilizar como glosario*
- **glossary_color** (*string*) – *Color de glosario*
- **repository_url** (*string*) – URL al estado del repositorio; vea *GET /api/components/(string:project)/(string:component)/repository/*
- **translations_url** (*string*) – URL a la lista de traducciones; vea *GET /api/components/(string:project)/(string:component)/translations/*
- **lock_url** (*string*) – URL to lock status; see *GET /api/components/(string:project)/(string:component)/lock/*
- **changes_list_url** (*string*) – URL to changes list; see *GET /api/components/(string:project)/(string:component)/changes/*
- **task_url** (*string*) – URL de una tarea en segundo plano (si corresponde); ver *GET /api/tasks/(str:uuid)/*

Example JSON data:


```
{
    "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***/**

Edit a component by a PATCH request.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **source_language** (*string*) – Código del idioma de origen del proyecto (opcional)

Objeto JSON de solicitud

- **name** (*string*) – nombre del componente
- **slug** (*string*) – slug of component
- **repo** (*string*) – VCS repository URL

CURL example:

```
curl \
    --data-binary '{"name": "new name"}' \
    -H "Content-Type: application/json" \
    -H "Authorization: Token TOKEN" \
    PATCH http://example.com/api/projects/hello/components/
```

JSON request example:

```
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"
}
```

Ejemplo de respuesta en JSON

```
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",
```

(continué en la próxima página)

(proviene de la página anterior)


```
            "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*)/

Edit a component by a PUT request.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de solicitud

- **branch** (*string*) – Rama del repositorio de control de versiones
- **file_format** (*string*) – formato de archivo de las traducciones
- **filemask** (*string*) – mask of translation files in the repository
- **name** (*string*) – nombre del componente
- **slug** (*string*) – slug of component
- **repo** (*string*) – VCS repository URL
- **template** (*string*) – base file for monolingual translations
- **new_base** (*string*) – base file for adding new translations
- **vcs** (*string*) – sistema de control de versiones

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

Nuevo en la versión 3.9.

Elimina un componente.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

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

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

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de respuesta

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

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

Nuevo en la versión 4.9.

Downloads all available translations associated with the component as an archive file using the requested format.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Parámetros de consulta

- **format** (*string*) – The archive format to use; If not specified, defaults to `zip`; Supported formats: `zip`

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

Devuelve un listado de las capturas de pantalla del componente.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de respuesta

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

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

Devuelve el estado de bloqueo del componente.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de respuesta

- **locked** (*boolean*) – whether component is locked for updates

Example JSON data:

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

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

Establece el estado de bloqueo del componente.

Response is same as *GET /api/components/(string:project)/(string:component)/lock/*.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de solicitud

- **lock** – Boolean whether to lock or not.

CURL example:

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

JSON request example:

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

{"lock": true}
```

Ejemplo de respuesta en JSON

```
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/

Returns information about VCS repository status.

The response is same as for *GET /api/projects/(string:project)/repository/*.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de respuesta

- **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*) – Remote commit information
- **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.

Parámetros

- **project** (*string*) – URL semántico del proyecto

- **component** (*string*) – URL semántico del componente

Objeto JSON de solicitud

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

Objeto JSON de respuesta

- **result** (*boolean*) – resultado de la operación

CURL example:

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

JSON request example:

```
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"}
```

Ejemplo de respuesta en JSON

```
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/**

Descarga el archivo de base para traducciones monolingües.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

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

Descarga el archivo de plantilla para traducciones nuevas.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

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

Devuelve un listado de objetos de traducción en el componente indicado.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de respuesta

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

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

Creates new translation in the given component.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de solicitud

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

Objeto JSON de respuesta

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

CURL example:

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

JSON request example:

```
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"}
```

Ejemplo de respuesta en JSON

```
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,
```

(continué en la próxima página)

(proviene de la página anterior)


```
    "have_suggestion": 0,
    "is_template": false,
    "is_source": false,
    "language": {
        "code": "cs",
        "direction": "ltr",
        "population": 1303174280
        "name": "Czech",
        "url": "http://example.com/api/languages/cs/",
        "web_url": "http://example.com/languages/cs/"
    },
    "language_code": "cs",
    "id": 125,
    "last_author": null,
    "last_change": null,
    "share_url": "http://example.com/engage/hello/cs/",
    "total": 4,
    "total_words": 15,
    "translate_url": "http://example.com/translate/hello/weblate/cs/",
    "translated": 0,
    "translated_percent": 0.0,
    "translated_words": 0,
    "url": "http://example.com/api/translations/hello/weblate/cs/",
    "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```


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

Devuelve estadísticas paginadas para todas las traducciones del componente.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Ver también:

Returned attributes are described in *Estadísticas*.

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

Devuelve proyectos vinculados con un componente.

Nuevo en la versión 4.5.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente

Objeto JSON de respuesta

- **projects** (*array*) – proyectos asociados; vea *GET /api/projects/(string:project)/*

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

Asociar un proyecto con un componente.

Nuevo en la versión 4.5.

Parámetros

- **project** (*string*) – URL semántico del proyecto

- **component** (*string*) – URL semántico del componente

Parámetros de Forma

- **string project_slug** – «Slug» del proyecto

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

Eliminar la asociación de un proyecto con un componente.

Nuevo en la versión 4.5.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **project_slug** (*string*) – Slug del proyecto a eliminar

1.12.9 Traducciones

GET /api/translations/

Devuelve un listado de traducciones.

Ver también:

Translation object attributes are documented at *GET /api/translations/(string:project)/(string:component)/(string:language)/*.

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

Devuelve información relativa a una traducción.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

Objeto JSON de respuesta

- **component** (*object*) – component object; see *GET /api/components/(string:project)/(string:component)/*
- **failing_checks** (*int*) – número de cadenas que fallan una comprobación
- **failing_checks_percent** (*float*) – porcentaje de cadenas que fallan una comprobación
- **failing_checks_words** (*int*) – número de palabras con comprobaciones fallidas
- **filename** (*string*) – nombre de archivo de la traducción
- **fuzzy** (*int*) – número de cadenas difusas (marcadas para editar)
- **fuzzy_percent** (*float*) – porcentaje de cadenas difusas (marcadas para editar)
- **fuzzy_words** (*int*) – número de palabras en cadenas difusas (marcadas para editar)
- **have_comment** (*int*) – número de cadenas con un comentario
- **have_suggestion** (*int*) – número de cadenas con una sugerencia
- **is_template** (*boolean*) – indica si la traducción tiene una base monolingüe
- **language** (*object*) – objeto de idioma de origen; vea *GET /api/languages/(string:language)/*

- **language_code** (*string*) – el código de idioma utilizado en el repositorio; puede ser distinto del código de idioma del objeto de idioma
- **last_author** (*string*) – nombre del último autor
- **last_change** (*timestamp*) – cronomarcador del último cambio
- **revision** (*string*) – revision hash for the file
- **share_url** (*string*) – URL para compartir que va a la página de participación
- **total** (*int*) – número total de cadenas
- **total_words** (*int*) – número total de palabras
- **translate_url** (*string*) – URL para traducir
- **translated** (*int*) – número de cadenas traducidas
- **translated_percent** (*float*) – porcentaje de cadenas traducidas
- **translated_words** (*int*) – número de palabras traducidas
- **repository_url** (*string*) – URL to repository status; see *GET /api/translations/(string:project)/(string:component)/(string:language)/repository/*
- **file_url** (*string*) – URL to file object; see *GET /api/translations/(string:project)/(string:component)/(string:language)/file/*
- **changes_list_url** (*string*) – URL to changes list; see *GET /api/translations/(string:project)/(string:component)/(string:language)/changes/*
- **units_list_url** (*string*) – URL to strings list; see *GET /api/translations/(string:project)/(string:component)/(string:language)/units/*

Example JSON data:


```
{
    "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",
```


(continué en la próxima página)

(proviene de la página anterior)


```
        "slug": "weblate",
        "template": "",
        "url": "http://example.com/api/components/hello/weblate/",
        "vcs": "git",
        "web_url": "http://example.com/projects/hello/weblate/"
    },
    "failing_checks": 3,
    "failing_checks_percent": 75.0,
    "failing_checks_words": 11,
    "filename": "po/cs.po",
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "fuzzy_words": 0,
    "have_comment": 0,
    "have_suggestion": 0,
    "is_template": false,
    "language": {
        "code": "cs",
        "direction": "ltr",
        "population": 1303174280
        "name": "Czech",
        "url": "http://example.com/api/languages/cs/",
        "web_url": "http://example.com/languages/cs/"
    },
    "language_code": "cs",
    "last_author": "Weblate Admin",
    "last_change": "2016-03-07T10:20:05.499",
    "revision": "7ddfafe6daaf57fc8654cc852ea6be212b015792",
    "share_url": "http://example.com/engage/hello/cs/",
    "total": 4,
    "total_words": 15,
    "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***/**

Nuevo en la versión 3.9.

Elimina una traducción.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

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

Returns a list of translation changes. This is essentially a translations-scoped *GET /api/changes/* accepting the same parameters.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

Objeto JSON de respuesta

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

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

Devuelve un listado de unidades de traducción.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción
- **q** (*string*) – Search query string *Búsquedas* (optional)

Objeto JSON de respuesta

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

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

Añadir nueva unidad.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

Objeto JSON de solicitud

- **key** (*string*) – Nombre de la unidad de traducción (utilizada como clave o contexto)
- **value** (*array*) – Source strings (use single string if not creating plural)
- **state** (*int*) – String state; see *GET /api/units/(int:id)/*

Objeto JSON de respuesta

- **unit** (*object*) – unidad recién creada; véase *GET /api/units/(int:id)/*

Ver también:

Gestionar cadenas, adding-new-strings

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

Trigger automatic translation.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

Objeto JSON de solicitud

- **mode** (*string*) – Modo de traducción automatizada
- **filter_type** (*string*) – Automatic translation filter type
- **auto_source** (*string*) – Fuente de traducción automática - `mt` o `others`

- **component** (*string*) – Permita que el proyecto contribuya a la memoria de traducción compartida para obtener acceso a componentes adicionales.
- **engines** (*array*) – Motores de traducción automática
- **threshold** (*string*) – Umbral de puntuación

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 *Descargar traducciones*).

Nota: 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.

Parámetros de consulta

- **format** – File format to use; if not specified no format conversion happens; see *Descargar traducciones* for supported formats
- **q** (*string*) – Filter downloaded strings, see search, only applicable when conversion is in place (`format` is specified).

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

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

Upload new file with translations.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

Parámetros de Forma

- **string conflicts** – How to deal with conflicts (`ignore`, `replace-translated` or `replace-approved`)
- **file file** – Archivo cargado
- **string email** – Correo electrónico del autor
- **string author** – Nombre del autor
- **string method** – Upload method (`translate`, `approve`, `suggest`, `fuzzy`, `replace`, `source`, `add`), see *Métodos de importación*
- **string fuzzy** – Fuzzy (marked for edit) strings processing (*empty*, `process`, `approve`)

CURL example:

```
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/**

Returns information about VCS repository status.

The response is same as for *GET /api/components/(string:project)/(string:component)/repository/*.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

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

Efectúa la operación indicada en el repositorio del sistema de control de versiones.

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

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

Objeto JSON de solicitud

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

Objeto JSON de respuesta

- **result** (*boolean*) – resultado de la operación

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

Devuelve estadísticas detalladas sobre la traducción.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **language** (*string*) – Código de idioma de la traducción

Ver también:

Returned attributes are described in *Estadísticas*.

1.12.10 Memoria

Nuevo en la versión 4.14.

GET /api/memory/

Devuelve una lista con los resultados de la memoria.

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

Elimina un objeto de la memoria

Parámetros

- **memory_object_id** – Memory Object ID

1.12.11 Unidades

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.

GET /api/units/

Devuelve un listado de unidades de traducción.

Parámetros

- **q** (*string*) – Search query string *Búsquedas* (optional)

Ver también:

Unit object attributes are documented at *GET /api/units/(int:id)/*.

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

Distinto en la versión 4.3: The `target` and `source` are now arrays to properly handle plural strings.

Returns information about translation unit.

Parámetros

- **id** (*int*) – Identificador de unidad

Objeto JSON de respuesta

- **translation** (*string*) – URL de un objeto de traducción relacionado
- **source** (*array*) – cadena de origen
- **previous_source** (*string*) – previous source string used for fuzzy matching
- **target** (*array*) – cadena de destino
- **id_hash** (*string*) – identificador único de la unidad
- **content_hash** (*string*) – identificador único de la cadena de origen
- **location** (*string*) – ubicación de la unidad en el código fuente
- **context** (*string*) – contexto de la unidad de traducción
- **note** (*string*) – nota de la unidad de traducción
- **flags** (*string*) – indicadores de la unidad de traducción
- **labels** (*array*) – translation unit labels, available on source units
- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved, 100 - read only
- **fuzzy** (*boolean*) – si la unidad está marcada como pendiente de trabajo o revisión
- **translated** (*boolean*) – indica si la unidad está traducida
- **approved** (*boolean*) – indica si la traducción está aprobada
- **position** (*int*) – posición de la unidad en el archivo de traducción
- **has_suggestion** (*boolean*) – indica si la unidad tiene sugerencias
- **has_comment** (*boolean*) – indica si la unidad tiene comentarios
- **has_failing_check** (*boolean*) – indica si la unidad tiene comprobaciones fallidas
- **num_words** (*int*) – número de palabras de origen
- **priority** (*int*) – prioridad de traducción; 100 es la predeterminada
- **id** (*int*) – identificador de unidad
- **explanation** (*string*) – String explanation, available on source units, see *Información adicional sobre las cadenas de origen*

- **extra_flags** (*string*) – Additional string flags, available on source units, see *Personalizar el comportamiento mediante indicadores*
- **web_url** (*string*) – URL para editar la unidad
- **source_unit** (*string*) – Enlace de la unidad fuente; ver *GET /api/units/(int:id)/*
- **pending** (*boolean*) – si la unidad está pendiente de escritura
- **timestamp** (*timestamp*) – Antigüedad de la cadena

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

Nuevo en la versión 4.3.

Realiza una actualización parcial de la unidad de la traducción.

Parámetros

- **id** (*int*) – Identificador de unidad

Objeto JSON de solicitud

- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see *Revisores dedicados*)
- **target** (*array*) – cadena de destino
- **explanation** (*string*) – String explanation, available on source units, see *Información adicional sobre las cadenas de origen*
- **extra_flags** (*string*) – Additional string flags, available on source units, see *Personalizar el comportamiento mediante indicadores*

Objeto JSON de respuesta

- **labels** (*array*) – labels, available on source units

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

Nuevo en la versión 4.3.

Realiza una actualización completa en la unidad de traducción.

Parámetros

- **id** (*int*) – Identificador de unidad

Objeto JSON de solicitud

- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see *Revisores dedicados*)
- **target** (*array*) – cadena de destino
- **explanation** (*string*) – String explanation, available on source units, see *Información adicional sobre las cadenas de origen*
- **extra_flags** (*string*) – Additional string flags, available on source units, see *Personalizar el comportamiento mediante indicadores*

Objeto JSON de respuesta

- **labels** (*array*) – labels, available on source units

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

Nuevo en la versión 4.3.

Elimina una unidad para la traducción.

Parámetros

- **id** (*int*) – Identificador de unidad

1.12.12 Cambios

GET /api/changes/

Distinto en la versión 4.1: Filtering of changes was introduced in the 4.1 release.

Devuelve un listado de cambios a la traducción.

Ver también:

Change object attributes are documented at *GET /api/changes/(int:id)/*.

Parámetros de consulta

- **user** (*string*) – Username of user to filters
- **action** (*int*) – Action to filter, can be used several times
- **timestamp_after** (*timestamp*) – ISO 8601 formatted timestamp to list changes after
- **timestamp_before** (*timestamp*) – ISO 8601 formatted timestamp to list changes before

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

Devuelve información relativa a un cambio de traducción.

Parámetros

- **id** (*int*) – Identificador de cambio

Objeto JSON de respuesta

- **unit** (*string*) – URL of a related unit object
- **translation** (*string*) – URL de un objeto de traducción relacionado
- **component** (*string*) – URL of a related component object
- **user** (*string*) – URL of a related user object
- **author** (*string*) – URL of a related author object
- **timestamp** (*timestamp*) – cronomarcador de suceso
- **action** (*int*) – identificación numérica de acción
- **action_name** (*string*) – descripción de texto de acción
- **target** (*string*) – event changed text or detail
- **id** (*int*) – change identifier

1.12.13 Capturas de pantalla

GET /api/screenshots/

Devuelve un listado de información de cadenas de capturas de pantalla.

Ver también:

Los atributos de los objetos de captura de pantalla están documentados en *GET /api/screenshots/(int:id)/*.

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

Devuelve información relativa a los datos de una captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla

Objeto JSON de respuesta

- **name** (*string*) – nombre de una captura de pantalla
- **component** (*string*) – URL of a related component object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Descargar la imagen de la captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla

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

Reemplazar la imagen de la captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla

Parámetros de Forma

- **file image** – Archivo cargado

CURL example:

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

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

Asociar la cadena de origen con una captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla

Parámetros de Forma

- **string unit_id** – Identificador de unidad

Objeto JSON de respuesta

- **name** (*string*) – nombre de una captura de pantalla
- **translation** (*string*) – URL de un objeto de traducción relacionado
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

int: *unit_id*

Quitar la asociación entre la cadena de origen y la captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla
- **unit_id** – Id. de unidad de cadena de origen

POST /api/screenshots/

Crea una captura de pantalla nueva.

Parámetros de Forma

- **file image** – Archivo cargado
- **string name** – Nombre de captura de pantalla
- **string project_slug** – «Slug» del proyecto
- **string component_slug** – «Slug» del componente
- **string language_code** – Código de idioma

Objeto JSON de respuesta

- **name** (*string*) – nombre de una captura de pantalla
- **component** (*string*) – URL of a related component object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Editar información parcial relativa a una captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla

Objeto JSON de respuesta

- **name** (*string*) – nombre de una captura de pantalla
- **component** (*string*) – URL of a related component object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Editar información completa relativa a una captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla

Objeto JSON de respuesta

- **name** (*string*) – nombre de una captura de pantalla
- **component** (*string*) – URL of a related component object
- **file_url** (*string*) – URL to download a file; see *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – link to associated source string information; see *GET /api/units/(int:id)/*

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

Eliminar captura de pantalla.

Parámetros

- **id** (*int*) – Identificador de captura de pantalla

1.12.14 Complementos

Nuevo en la versión 4.4.1.

GET /api/addons/

Devuelve un listado de complementos.

Ver también:

Los atributos de objeto de complemento se documentan en *GET /api/addons/(int:id)/*.

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

Devuelve información relativa a los datos de un complemento.

Parámetros

- **id** (*int*) – ID del complemento

Objeto JSON de respuesta

- **name** (*string*) – nombre de un complemento
- **component** (*string*) – URL of a related component object
- **configuration** (*object*) – Configuración opcional del complemento

Ver también:

Complementos

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

Crea un complemento nuevo.

Parámetros

- **project_slug** (*string*) – «Slug» del proyecto
- **component_slug** (*string*) – «Slug» del componente

Objeto JSON de solicitud

- **name** (*string*) – nombre de un complemento
- **configuration** (*object*) – Configuración opcional del complemento

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

Editar información parcial relativa a un complemento.

Parámetros

- **id** (*int*) – ID del complemento

Objeto JSON de respuesta

- **configuration** (*object*) – Configuración opcional del complemento

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

Editar información completa relativa a un complemento.

Parámetros

- **id** (*int*) – ID del complemento

Objeto JSON de respuesta

- **configuration** (*object*) – Configuración opcional del complemento

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

Eliminar el complemento.

Parámetros

- **id** (*int*) – ID del complemento

1.12.15 Listas de componentes

Nuevo en la versión 4.0.

GET /api/component-lists/

Devuelve un listado de listas de componentes.

Ver también:

Component list object attributes are documented at *GET /api/component-lists/(str:slug)/*.

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

Devuelve información relativa a la lista de componentes.

Parámetros

- **slug** (*string*) – Component list slug

Objeto JSON de respuesta

- **name** (*string*) – nombre de una lista de componentes
- **slug** (*string*) – slug of a component list
- **show_dashboard** (*boolean*) – whether to show it on a dashboard
- **components** (*array*) – link to associated components; see *GET /api/components/(string:project)/(string:component)/*
- **auto_assign** (*array*) – reglas de asignación automáticas

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

Cambia los parámetros de la lista de componentes.

Parámetros

- **slug** (*string*) – Component list slug

Objeto JSON de solicitud

- **name** (*string*) – nombre de una lista de componentes
- **slug** (*string*) – slug of a component list
- **show_dashboard** (*boolean*) – whether to show it on a dashboard

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

Cambia los parámetros de la lista de componentes.

Parámetros

- **slug** (*string*) – Component list slug

Objeto JSON de solicitud

- **name** (*string*) – nombre de una lista de componentes
- **slug** (*string*) – slug of a component list
- **show_dashboard** (*boolean*) – whether to show it on a dashboard

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

Elimina la lista de componentes.

Parámetros

- **slug** (*string*) – Component list slug

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

Associate component with a component list.

Parámetros

- **slug** (*string*) – Component list slug

Parámetros de Forma

- **string component_id** – Identificador de componente

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

Disassociate a component from the component list.

Parámetros

- **slug** (*string*) – Component list slug
- **component_slug** (*string*) – «Slug» del componente

1.12.16 Glosario

Distinto en la versión 4.5: Glossaries are now stored as regular components, translations and strings, please use respective API instead.

1.12.17 Tareas

Nuevo en la versión 4.4.

GET /api/tasks/

La enumeración de las tareas no está disponible actualmente.

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

Devuelve información relativa a una tarea

Parámetros

- **uuid** (*string*) – UUID de la tarea

Objeto JSON de respuesta

- **completed** (*boolean*) – Indica si la tarea se ha completado
- **progress** (*int*) – Progreso de la tarea expresado en porcentaje
- **result** (*object*) – Resultado de la tarea o detalles de progreso
- **log** (*string*) – Registro de tareas

1.12.18 Estadísticas

GET /api/(**str:** *object*)**/statistics/**

There are several statistics endpoints for objects and all of them contain same structure.

Parámetros

- **object** (*string*) – URL path

Objeto JSON de respuesta

- **total** (*int*) – número total de cadenas
- **total_words** (*int*) – número total de palabras
- **total_chars** (*int*) – total number of characters
- **last_change** (*timestamp*) – fecha del último cambio
- **translated** (*int*) – número de cadenas traducidas
- **translated_percent** (*float*) – porcentaje de cadenas traducidas
- **translated_words** (*int*) – número de palabras traducidas
- **translated_words_percent** (*float*) – porcentaje de palabras traducidas
- **translated_chars** (*int*) – número de caracteres traducidos
- **translated_chars_percent** (*float*) – porcentaje de caracteres traducidos
- **fuzzy** (*int*) – número de cadenas difusas (marcadas para editar)
- **fuzzy_percent** (*float*) – porcentaje de cadenas difusas (marcadas para editar)
- **failing** (*int*) – número de comprobaciones fallidas
- **failing_percent** (*float*) – porcentaje de comprobaciones fallidas
- **approved** (*int*) – number of approved checks
- **approved_percent** (*float*) – percentage of approved strings
- **readonly** (*int*) – number of read-only strings
- **readonly_percent** (*float*) – percentage of read-only strings
- **suggestions** (*int*) – number of strings with suggestions
- **comments** (*int*) – number of strings with comments
- **name** (*string*) – object name
- **url** (*string*) – URL to access the object (if applicable)
- **url_translate** (*string*) – URL to access the translation (if applicable)
- **code** (*string*) – language code (if applicable)

Ver también:

GET /api/languages/(string:language)/statistics/, *GET /api/projects/(string:project)/statistics/*, *GET /api/components/(string:project)/(string:component)/statistics/*, *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/*

1.12.19 Métrica

GET /api/metrics/

Returns server metrics.

Objeto JSON de respuesta

- **units** (*int*) – Número de unidades
- **units_translated** (*int*) – Número de unidades traducidas
- **users** (*int*) – Numero de usuarios
- **changes** (*int*) – Cantidad de cambios
- **projects** (*int*) – Número de proyectos
- **components** (*int*) – Número de componentes
- **translations** (*int*) – Número de traducciones
- **languages** (*int*) – Número de idiomas utilizados
- **checks** (*int*) – Número de los controles de calidad activados
- **configuration_errors** (*int*) – Número de errores en la configuración
- **suggestions** (*int*) – Número de sugerencias pendientes
- **celery_queues** (*object*) – Longitudes de las colas de Celery, véase *Tareas en segundo plano con Celery*.
- **name** (*string*) – Nombre del servidor configurado

1.12.20 Buscar

GET /api/search/

Nuevo en la versión 4.18.

Returns site-wide search results as a list. There is no pagination on the result set, only first few matches are returned for each category.

Objeto JSON de respuesta

- **name** (*str*) – Name of the matched item.
- **url** (*str*) – Web URL of the matched item.
- **category** (*str*) – Category of the matched item.

1.12.21 Actuadores de notificación

Los actuadores de notificación permiten a aplicaciones externas notificar a Weblate de que se ha actualizado un repositorio de control de versiones.

You can use repository endpoints for projects, components and translations to update individual repositories; see *POST /api/projects/(string:project)/repository/* for documentation.

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

Obsoleto desde la versión 2.6: Please use *POST /api/components/(string:project)/(string:component)/repository/* instead which works properly with authentication for ACL limited projects.

Triggers update of a component (pulling from VCS and scanning for translation changes).

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

Obsoleto desde la versión 2.6: Please use *POST /api/projects/(string:project)/repository/* instead which works properly with authentication for ACL limited projects.

Triggers update of all components in a project (pulling from VCS and scanning for translation changes).

POST /hooks/github/

Special hook for handling GitHub notifications and automatically updating matching components.

Nota: GitHub includes direct support for notifying Weblate: enable Weblate service hook in repository settings and set the URL to the URL of your Weblate installation.

Ver también:

Recibir cambios automáticamente de GitHub
For instruction on setting up GitHub integration

https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks
Información genérica sobre los actuadores web de GitHub

ENABLE_HOOKS
Para activar actuadores en todo Weblate

POST /hooks/gitlab/

Special hook for handling GitLab notifications and automatically updating matching components.

Ver también:

Automatically receiving changes from GitLab
For instruction on setting up GitLab integration

https://docs.gitlab.com/ee/user/project/integrations/webhooks.html
Información genérica sobre los actuadores web de GitLab

ENABLE_HOOKS
Para activar actuadores en todo Weblate

POST /hooks/bitbucket/

Special hook for handling Bitbucket notifications and automatically updating matching components.

Ver también:

Recibir cambios automáticamente de Bitbucket
For instruction on setting up Bitbucket integration

https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/
Información genérica sobre los actuadores web de Bitbucket

ENABLE_HOOKS
Para activar actuadores en todo Weblate

POST /hooks/pagure/

Nuevo en la versión 3.3.

Special hook for handling Pagure notifications and automatically updating matching components.

Ver también:

Recibir cambios automáticamente de Pagure
For instruction on setting up Pagure integration

https://docs.pagure.org/pagure/usage/using_webhooks.html
Información genérica sobre los actuadores web de Pagure

ENABLE_HOOKS
Para activar actuadores en todo Weblate

POST /hooks/azure/

Nuevo en la versión 3.8.

Actuador especial para trabajar con las notificaciones de Azure DevOps y actualizar automáticamente los componentes que correspondan.

Nota: Cerciórese de que *Detalles de recursos que enviar* esté configurada como *Todo*; de lo contrario, Weblate no podrá encontrar su repositorio de Azure.

Ver también:

Recibir cambios automáticamente de Azure Repos
For instruction on setting up Azure integration

https://learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops
Información genérica sobre los actuadores web de Azure DevOps

ENABLE_HOOKS
Para activar actuadores en todo Weblate

POST /hooks/gitea/

Nuevo en la versión 3.9.

Special hook for handling Gitea Webhook notifications and automatically updating matching components.

Ver también:

Recibir cambios automáticamente de Gitea
For instruction on setting up Gitea integration

https://docs.gitea.io/en-us/webhooks/
Información genérica sobre los actuadores web de Gitea

ENABLE_HOOKS
Para activar actuadores en todo Weblate

POST /hooks/gitee/

Nuevo en la versión 3.9.

Special hook for handling Gitee Webhook notifications and automatically updating matching components.

Ver también:

Recibir cambios automáticamente de Gitee
For instruction on setting up Gitee integration

https://gitee.com/help/categories/40
Información genérica sobre los actuadores web de Gitee

ENABLE_HOOKS
Para activar actuadores en todo Weblate

1.12.22 Exportaciones

Weblate brinda diversas exportaciones para permitirle un tratamiento posterior de sus datos.

GET /exports/stats/(string: *project*)/
string: *component*/

Parámetros de consulta

- **format** (*string*) – Formato de salida: `json` o `csv`

Obsoleto desde la versión 2.6: Utilice *GET /api/components/(string:project)/(string:component)/statistics/* y *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/* en su lugar; esto permite además el acceso a los proyectos controlados por ACL.

Recupera las estadísticas del componente indicado en el formato que se indique.

Example request:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

[
    {
        "code": "cs",
        "failing": 0,
        "failing_percent": 0.0,
        "fuzzy": 0,
        "fuzzy_percent": 0.0,
        "last_author": "Michal Čihař",
        "last_change": "2012-03-28T15:07:38+00:00",
        "name": "Czech",
        "total": 436,
        "total_words": 15271,
        "translated": 436,
        "translated_percent": 100.0,
        "translated_words": 3201,
        "url": "http://hosted.weblate.org/engage/weblate/cs/",
        "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
    },
    {
        "code": "nl",
        "failing": 21,
        "failing_percent": 4.8,
        "fuzzy": 11,
        "fuzzy_percent": 2.5,
        "last_author": null,
        "last_change": null,
        "name": "Dutch",
        "total": 436,
        "total_words": 15271,
        "translated": 319,
        "translated_percent": 73.2,
        "translated_words": 3201,
        "url": "http://hosted.weblate.org/engage/weblate/nl/",
        "url_translate": "http://hosted.weblate.org/projects/weblate/main/nl/"
```

(continué en la próxima página)

(proviene de la página anterior)


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


1.12.23 Sindicación RSS

Las modificaciones a las traducciones se exportan en suministros RSS.

GET /exports/rss/(**string:** *project*)**/**
string: *component***/string:** *language***/**

Recupera el suministro RSS de los cambios recientes de una traducción.

GET /exports/rss/(**string:** *project*)**/**
string: *component***/**

Recupera el suministro RSS de los cambios recientes de un componente.

GET /exports/rss/(**string:** *project*)**/**

Recupera el suministro RSS de los cambios recientes de un proyecto.

GET /exports/rss/language/(**string:** *language*)**/**

Recupera el suministro RSS de los cambios recientes de un idioma.

GET /exports/rss/

Recupera el suministro RSS de los cambios recientes de la instalación de Weblate.

Ver también:

RSS on Wikipedia

1.13 Cliente de Weblate

1.13.1 Instalación

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 Uso de Docker

El cliente de Weblate está disponible también como imagen para Docker.

The image is published on Docker Hub: https://hub.docker.com/r/weblate/wlc

Instalación:

```
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]
```

Ejemplo:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

You might want to pass your *Archivos de configuración* 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 Primeros pasos

The wlc configuration is stored in `~/.config/weblate` (see *Archivos de configuración* 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
```

Ver también:

Archivos de configuración

1.13.4 Synopsis

```
wlc [arguments] <command> [options]
```

Las órdenes indican cuál operación ha de realizarse.

1.13.5 Descripción

Weblate Client is a Python library and command-line utility to manage Weblate remotely using *API REST de Weblate*. The command-line utility can be invoked as **wlc** and is built-in on `wlc`.

Argumentos

El programa acepta los argumentos siguientes, que definen el formato de la salida, o bien, cuál instalación de Weblate utilizar. Deben introducirse antes de cualquier orden.

--format `{csv,json,text,html}`

Especifique el formato de la salida.

--url `URL`

Specify the API URL. Overrides any value found in the configuration file, see *Archivos de configuración*. The URL should end with `/api/`, for example `https://hosted.weblate.org/api/`.

--key `KEY`

Specify the API user key to use. Overrides any value found in the configuration file, see *Archivos de configuración*. You can find your key in your profile on Weblate.

--config `PATH`

Overrides the configuration file path, see *Archivos de configuración*.

--config-section `SECTION`

Overrides configuration file section in use, see *Archivos de configuración*.

Órdenes

Las siguientes órdenes están disponibles:

version

Muestra la versión actual.

list-languages

Enumera los idiomas utilizados en Weblate.

list-projects

Enumera los proyectos en Weblate.

list-components

Enumera los componentes en Weblate.

list-translations

Enumera las traducciones en Weblate.

show

Muestra un objeto de Weblate (traducción, componente o proyecto).

ls

Enumera objetos de Weblate (traducción, componente o proyecto).

commit

Consigna los cambios efectuados en un objeto de Weblate (traducción, componente o proyecto).

pull

Incorpora los cambios en el repositorio remoto en un objeto de Weblate (traducción, componente o proyecto).

push

Envía los cambios en el objeto de Weblate (traducción, componente o proyecto) al repositorio remoto.

reset

Nuevo en la versión 0.7: Admitido desde wlc 0.7.

Restablece los cambios en un objeto de Weblate (traducción, componente o proyecto) para que coincida con el repositorio remoto.

cleanup

Nuevo en la versión 0.9: Admitido desde wlc 0.9.

Removes any untracked changes in a Weblate object to match the remote repository (translation, component or project).

repo

Displays repository status for a given Weblate object (translation, component or project).

stats

Displays detailed statistics for a given Weblate object (translation, component or project).

lock-status

Nuevo en la versión 0.5: Admitido desde wlc 0.5.

Muestra el estado de bloqueo.

lock

Nuevo en la versión 0.5: Admitido desde wlc 0.5.

Bloquea el componente para evitar más traducciones en Weblate.

unlock

Nuevo en la versión 0.5: Admitido desde wlc 0.5.

Desbloquea la traducción del componente de Weblate.

changes

Nuevo en la versión 0.7: Admitido desde wlc 0.7 y Weblate 2.10.

Muestra los cambios correspondientes al objeto indicado.

download

Nuevo en la versión 0.7: Admitido desde wlc 0.7.

Descarga un archivo de traducción.

--convert

Convierte el formato de archivo. Si no se especifica ningún formato, no se produce ninguna conversión en el servidor y el archivo se descarga tal cual es en el repositorio.

--output

Permite especificar el archivo en el que se guardará la salida; si no se define, se mostrará en la salida estándar.

upload

Nuevo en la versión 0.9: Admitido desde wlc 0.9.

Carga un archivo de traducción.

--overwrite

Sobrescribir traducciones existentes al cargar.

--input

Archivo del cual se lee contenido; si no se define, se lee de la entrada estándar.

--method

Upload method to use, see *Métodos de importación*.

--fuzzy
Fuzzy (marked for edit) strings processing (*empty*, `process`, `approve`)

--author-name
Author name, to override currently authenticated user

--author-email
Author e-mail, to override currently authenticated user

Consejo: Puede obtener más información al invocar cada orden si añade `--help`; por ejemplo: `wlc ls --help`.

1.13.6 Archivos de configuración

.weblate, .weblate.ini, weblate.ini
Archivo de configuración por proyecto

C:\Users\NOMBRE\AppData\weblate.ini
Archivo de configuración de usuario en Windows.

~/.config/weblate
Archivo de configuración de usuario

/etc/xdg/weblate
Archivo de configuración para todo el sistema

El programa sigue la especificación XDG, de modo que puede ajustar la colocación de los archivos de configuración mediante las variables de entorno `XDG_CONFIG_HOME` o `XDG_CONFIG_DIRS`. En Windows, se prefiere el directorio `APPDATA` para ubicar el archivo de configuración.

Following settings can be configured in the `[weblate]` section (you can customize this by *`--config-section`*):

key
Clave de API para obtener acceso a Weblate.

url
URL del servidor de la API; el valor predeterminado es `http://127.0.0.1:8000/api/`.

translation
Path to the default translation - component or project.

El archivo de configuración es un INI; por ejemplo:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Además, las claves de la API se pueden almacenar en la sección `[keys]`:

```
[keys]
https://hosted.weblate.org/api/ = APIKEY
```

This allows you to store keys in your personal settings, while using the `.weblate` configuration in the VCS repository so that wlc knows which server it should talk to.

1.13.7 Ejemplos

Mostrar la versión actual del programa:

```
$ wlc version
version: 0.1
```

Enumerar todos los proyectos:

```
$ 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/
```

Cargar un archivo de traducción:

```
$ wlc upload project/component/language --input /tmp/hello.po
```

También puede designar el proyecto en el que debe funcionar wlc:

```
$ 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/
```

Con esta configuración es sencillo consignar los cambios pendientes del proyecto actual:

```
$ wlc commit
```

1.14 API de Python de Weblate

1.14.1 Instalación

The Python API is shipped separately, you need to install the *Cliente de Weblate* (wlc) to have it.

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception wlc.**WeblateException**

Base class for all exceptions.

Weblate

class wlc.**Weblate**(*key=''*, *url=None*, *config=None*)

Parámetros

- **key** (*str*) – User key
- **url** (*str*) – API server URL, if not specified default is used
- **config** (wlc.config.WeblateConfig) – Configuration object, overrides any other parameters.

Access class to the API, define API key and optionally API URL.

get(*path*)

Parámetros
path (*str*) – Request path

Tipo del valor devuelto
object

Performs a single API GET call.

post(*path*, ***kwargs*)

Parámetros
path (*str*) – Request path

Tipo del valor devuelto
object

Performs a single API GET call.

1.14.3 wlc.config

WeblateConfig

class wlc.config.**WeblateConfig**(*section='wlc'*)

Parámetros
section (*str*) – Configuration section to use

Configuration file parser following XDG specification.

load(*path=None*)

Parámetros
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*)

Parámetros

- **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*)

Clase principal para invocar órdenes.

CAPÍTULO 2

Documentación para administradores

2.1 Instrucciones de configuración

2.1.1 Instalar Weblate

Instalar con Docker

With dockerized Weblate deployment you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database.

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espacio de almacenamiento

Cuanta más memoria tenga, mejor, ya que se utiliza para el prealmacenaje en todos los niveles (sistema de archivos, base de datos y Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

Nota: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

Instalación

The following examples assume you have a working Docker environment, with `docker-compose-plugin` installed. Please check the Docker documentation for instructions.

1. Clone el repositorio weblate-docker:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Create a `docker-compose.override.yml` file with your settings. See *Docker environment variables* for full list of environment variables.

```
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
```

Nota: If *WEBLATE_ADMIN_PASSWORD* is not set, the admin user is created with a random password shown on first startup.

The provided example makes Weblate listen on port 80, edit the port mapping in the `docker-compose.override.yml` file to change it.

3. Inicie los contenedores de Weblate:

```
docker compose up
```

Enjoy your Weblate deployment, it's accessible on port 80 of the `weblate` container.

Distinto en la versión 3.7.1-6: In July 2019 (starting with the 3.7.1-6 tag), the containers are not running as a root user. This has changed the exposed port from 80 to 8080.

Ver también:

Invocar órdenes de gestión

Choosing Docker image registry

Weblate containers are published to following registries:

- Docker Hub, see https://hub.docker.com/r/weblate/weblate
- GitHub Packages registry, see https://github.com/WeblateOrg/docker/pkgs/container/weblate

Nota: All examples currently fetch images from Docker Hub, please adjust the configuration accordingly to use a different registry.

Choosing Docker image tag

Please choose a tag that matches your environment and expectations:

Nombre de la etiqueta	Descripción	Caso de uso
`latest`	Weblate stable release, matches latest tagged release	Rolling updates in a production environment
`<MAJOR>`	Versión estable de Weblate	Rolling updates within a major version in a production environment
`<MAJOR>.<MINOR>`	Versión estable de Weblate	Rolling updates within a minor version in a production environment
`<VERSION>.<PATCH>`	Versión estable de Weblate	Well defined deploy in a production environment
`edge`	Weblate stable release with development changes in the Docker container (for example updated dependencies)	Rolling updates in a staging environment
`edge-<DATE>-`	Weblate stable release with development changes in the Docker container (for example updated dependencies)	Well defined deploy in a staging environment
`bleeding`	Development version Weblate from Git	Rollling updates to test upcoming Weblate features
`bleeding-<DA`	Development version Weblate from Git	Well defined deploy to test upcoming Weblate features

Every image is tested by our CI before it gets published, so even the *bleeding* version should be quite safe to use.

Full list of published tags can be found at GitHub Packages

Contenedor Docker con compatibilidad con HTTPS

Please see *Instalación* for generic deployment instructions, this section only mentions differences compared to it.

Utilizar certificados SSL propios

Nuevo en la versión 3.8-3.

In case you have own SSL certificate you want to use, simply place the files into the Weblate data volume (see *Volúmenes de contenedores Docker*):

- `ssl/fullchain.pem`, que contiene el certificado SSL y cualquier certificado CA que se necesite
- `ssl/privkey.pem`, que contiene la clave privada

Both of these files must be owned by the same user as the one starting the docker container and have file mask set to `600` (readable and writable only by the owning user).

Additionally, Weblate container will now accept SSL connections on port 4443, you will want to include the port forwarding for HTTPS in docker compose override:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

If you already host other sites on the same server, it is likely ports `80` and `443` are used by a reverse proxy, such as NGINX. To pass the HTTPS connection from NGINX to the docker container, you can use the following configuration:

```
server {
    listen 443 ssl;
    listen [::]:443 ssl;

    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>;
    }
}
```

Replace `<SITE_URL>`, `<SITE>` and `<EXPOSED_DOCKER_PORT>` with actual values from your environment.

Certificados SSL automáticos con Let's Encrypt

In case you want to use Let's Encrypt automatically generated SSL certificates on public installation, you need to add a reverse HTTPS proxy an additional Docker container, https-portal will be used for that. This is made use of in the `docker-compose-https.yml` file. Then create a `docker-compose-https.override.yml` file with your settings:

```
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'
```

Whenever invoking **`docker compose`** you need to pass both files to it, and then do:

```
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
```

Actualizar el contenedor de Docker

Usually it is good idea to only update the Weblate container and keep the PostgreSQL container at the version you have, as upgrading PostgreSQL is quite painful and in most cases does not bring many benefits.

Distinto en la versión 4.17-1: Since Weblate 4.17-1, the Docker container uses Django 4.2 what requires PostgreSQL 12 or newer, please upgrade it prior to upgrading Weblate. See *Actualización del contenedor de PostgreSQL*.

You can do this by sticking with the existing docker-compose and just pull the latest images and then restart:

```
# 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
```

The Weblate database should be automatically migrated on first startup, and there should be no need for additional manual actions.

Nota: Upgrades across major versions are not supported by Weblate. For example, if you are on 3.x series and want to upgrade to 4.x, first upgrade to the latest 4.0.x-y image (at time of writing this it is the `4.0.4-5`), which will do the migration and then continue upgrading to newer versions.

You might also want to update the `docker-compose` repository, though it's not needed in most case. See *Actualización del contenedor de PostgreSQL* for upgrading the PostgreSQL server.

Actualización del contenedor de PostgreSQL

PostgreSQL containers do not support automatic upgrading between version, you need to perform the upgrade manually. Following steps show one of the options of upgrading.

Ver también:

https://github.com/docker-library/postgres/issues/37

1. Detener el contenedor de Weblate:

   ```
   docker compose stop weblate cache
   ```

2. Backup the database:

   ```
   docker compose exec database pg_dumpall --clean --if-exists --username weblate␣
   ↪> backup.sql
   ```

3. Detenga el contenedor de la base de datos:

   ```
   docker compose stop database
   ```

4. Elimine el volumen PostgreSQL:

   ```
   docker compose rm -v database
   docker volume remove weblate-docker_postgres-data
   ```

5. Adjust `docker-compose.yml` to use new PostgreSQL version.

6. Inicie el contenedor de la base de datos:

```
docker compose up -d database
```

7. Restaurar la base de datos a partir de la copia de seguridad:

```
cat backup.sql | docker compose exec -T database psql --username weblate --
↪dbname weblate
```

Consejo: Please check that the database name matches *POSTGRES_DATABASE*.

8. (Optional) Update password for the Weblate user. This might be needed when migrating to PostgreSQL 14 or 15 as way of storing passwords has been changed:

```
docker compose exec -T database psql --username weblate --dbname weblate -c
↪"ALTER USER weblate WITH PASSWORD 'weblate'"
```

Consejo: Please check that the database name matches *POSTGRES_DATABASE*.

9. Inicie todos los contenedores restantes:

```
docker compose up -d
```

Iniciar la sesión cómo administrador

After container setup, you can sign in as *admin* user with password provided in *WEBLATE_ADMIN_PASSWORD*, or a random password generated on first start if that was not set.

To reset *admin* password, restart the container with *WEBLATE_ADMIN_PASSWORD* set to new password.

Ver también:

WEBLATE_ADMIN_PASSWORD, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

Number of processes and memory consumption

The number of worker processes for both uWSGI and Celery is determined automatically based on number of CPUs. This works well for most cloud virtual machines as these typically have few CPUs and good amount of memory.

In case you have a lot of CPU cores and hit out of memory issues, try reducing number of workers:

```
environment:
  WEBLATE_WORKERS: 2
```

También puede ajustar las categorías individuales de los trabajadores:

```
environment:
  WEB_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Ver también:

WEBLATE_WORKERS, *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS*, *WEB_WORKERS*

Scaling horizontally

Nuevo en la versión 4.6.

You can run multiple Weblate containers to scale the service horizontally. The `/app/data` volume has to be shared by all containers, it is recommended to use cluster filesystem such as GlusterFS for this. The `/app/cache` volume should be separate for each container.

Each Weblate container has defined role using *WEBLATE_SERVICE* environment variable. Please follow carefully the documentation as some of the services should be running just once in the cluster and the ordering of the services matters as well.

You can find example setup in the `docker-compose` repo as docker-compose-split.yml.

Docker environment variables

Many of Weblate's *Configuración* can be set in the Docker container using the environment variables described below.

If you need to define a setting not exposed through Docker environment variables, see *Configuration beyond environment variables*.

Generic settings

WEBLATE_DEBUG

Configures Django debug mode using *DEBUG*.

Ejemplo:

```
environment:
  WEBLATE_DEBUG: 1
```

Ver también:

Desactivar el modo de depuración

WEBLATE_LOGLEVEL

Configures the logging verbosity. Set this to `DEBUG` to get more detailed logs.

Defaults to `INFO` when *WEBLATE_DEBUG* is turned off, `DEBUG` is used when debug mode is turned on.

WEBLATE_LOGLEVEL_DATABASE

Configures the logging of the database queries verbosity.

WEBLATE_SITE_TITLE

Modifica el título del sitio que se muestra en la cabecera de todas las páginas.

WEBLATE_SITE_DOMAIN

Configura el dominio del sitio. Este parámetro es obligatorio.

Ver también:

Set correct site domain, *SITE_DOMAIN*

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Configures the site-admin's name and e-mail. It is used for both *ADMINS* setting and creating *admin* user (see *WEBLATE_ADMIN_PASSWORD* for more info on that).

Ejemplo:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

Ver también:

Iniciar la sesión cómo administrador, *Properly configure admins*, *ADMINS*

WEBLATE_ADMIN_PASSWORD

Sets the password for the *admin* user.

- If not set and *admin* user does not exist, it is created with a random password shown on first container startup.
- If not set and *admin* user exists, no action is performed.
- If set the *admin* user is adjusted on every container startup to match *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_NAME* and *WEBLATE_ADMIN_EMAIL*.

Advertencia: It might be a security risk to store password in the configuration file. Consider using this variable only for initial setup (or let Weblate generate random password on initial startup) or for password recovery.

Ver también:

Iniciar la sesión cómo administrador, *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_PASSWORD_FILE*, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

WEBLATE_ADMIN_PASSWORD_FILE

Sets the path to a file containing the password for the *admin* user.

Ver también:

WEBLATE_ADMIN_PASSWORD

WEBLATE_SERVER_EMAIL

The email address that error messages are sent from.

Ver también:

SERVER_EMAIL, *Configure e-mail sending*

WEBLATE_DEFAULT_FROM_EMAIL

Configures the address for outgoing e-mails.

Ver también:

DEFAULT_FROM_EMAIL, *Configure e-mail sending*

WEBLATE_CONTACT_FORM

Configures contact form behavior, see *CONTACT_FORM*.

WEBLATE_ALLOWED_HOSTS

Configures allowed HTTP hostnames using *ALLOWED_HOSTS*.

El valor predeterminado es `*`, que permite todos los nombres de anfitrión.

Ejemplo:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com,example.com
```

Ver también:

ALLOWED_HOSTS, *Allowed hosts setup*, *Set correct site domain*

WEBLATE_REGISTRATION_OPEN

Configures whether registrations are open by toggling *REGISTRATION_OPEN*.

Ejemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Configure which authentication methods can be used to create new account via *REGISTRATION_ALLOW_BACKENDS*.

Ejemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
↪oauth2
```

WEBLATE_REGISTRATION_REBIND

Nuevo en la versión 4.16.

Configures *REGISTRATION_REBIND*.

WEBLATE_TIME_ZONE

Configura el huso horario utilizado en Weblate; vea TIME_ZONE.

Nota: Para cambiar el huso horario del contenedor Docker, utilice la variable de entorno TZ.

Ejemplo:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Makes Weblate assume it is operated behind a reverse HTTPS proxy, it makes Weblate use HTTPS in e-mail and API links or set secure flags on cookies.

Consejo: Please see *ENABLE_HTTPS* documentation for possible caveats.

Nota: Esto no hace que el contenedor de Weblate acepte las conexiones HTTPS; debe configurarlas también. Vea *Contenedor Docker con compatibilidad con HTTPS* para obtener ejemplos.

Ejemplo:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Ver también:

ENABLE_HTTPS Set correct site domain, *WEBLATE_SECURE_PROXY_SSL_HEADER*

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

Nuevo en la versión 4.12.1.

Lets Weblate set the *meta[name=monetization]* field in the head of the document. If multiple are specified, chooses one randomly.

Ver también:

INTERLEDGER_PAYMENT_POINTERS

WEBLATE_IP_PROXY_HEADER

Permite que Weblate recupere la dirección IP de cualquier cabecera HTTP que se indique. Utilice esta variable si usa un «proxy» inverso ante el contenedor de Weblate.

Enables *IP_BEHIND_REVERSE_PROXY* and sets *IP_PROXY_HEADER*.

Nota: The format must conform to Django's expectations. Django transforms raw HTTP header names as follows:

- convierte todas las letras en mayúsculas
- sustituye cualquier guion por guiones bajos
- antepone el prefijo `HTTP_`

So `X-Forwarded-For` would be mapped to `HTTP_X_FORWARDED_FOR`.

Ejemplo:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

A tuple representing a HTTP header/value combination that signifies a request is secure. This is needed when Weblate is running behind a reverse proxy doing SSL termination which does not pass standard HTTPS headers.

Ejemplo:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

Ver también:

SECURE_PROXY_SSL_HEADER

WEBLATE_REQUIRE_LOGIN

Enables *REQUIRE_LOGIN* to enforce authentication on whole Weblate.

Ejemplo:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Adds URL exceptions for authentication required for the whole Weblate installation using *LOGIN_REQUIRED_URLS_EXCEPTIONS*.

You can either replace whole settings, or modify default value using `ADD` and `REMOVE` variables.

WEBLATE_GOOGLE_ANALYTICS_ID

Configures ID for Google Analytics by changing *GOOGLE_ANALYTICS_ID*.

WEBLATE_GITHUB_USERNAME

WEBLATE_GITHUB_TOKEN

WEBLATE_GITHUB_HOST

Configures GitHub pull-requests integration by changing *GITHUB_CREDENTIALS*.

Ver también:

Solicitudes de incorporación de GitHub

WEBLATE_GITLAB_USERNAME

WEBLATE_GITLAB_TOKEN

WEBLATE_GITLAB_HOST

Configures GitLab merge-requests integration by changing *GITLAB_CREDENTIALS*.

Ejemplo:

```
WEBLATE_GITLAB_USERNAME=weblate
WEBLATE_GITLAB_HOST=gitlab.com
WEBLATE_GITLAB_TOKEN=token
```

Ver también:

Solicitudes de fusión de GitLab

WEBLATE_GITEA_USERNAME

WEBLATE_GITEA_TOKEN

WEBLATE_GITEA_HOST

Configures Gitea pull-requests integration by changing *GITEA_CREDENTIALS*.

Ver también:

Solicitudes de incorporación de Gitea

WEBLATE_PAGURE_USERNAME

WEBLATE_PAGURE_TOKEN

WEBLATE_PAGURE_HOST

Configures Pagure merge-requests integration by changing *PAGURE_CREDENTIALS*.

Ver también:

Solicitudes de fusión de Pagure

WEBLATE_BITBUCKETSERVER_USERNAME

WEBLATE_BITBUCKETSERVER_TOKEN

WEBLATE_BITBUCKETSERVER_HOST

Configures Bitbucket Server pull-requests integration by changing *BITBUCKETSERVER_CREDENTIALS*.

Ver también:

Bitbucket Server pull requests

WEBLATE_DEFAULT_PULL_MESSAGE

Configures the default title and message for pull requests via API by changing *DEFAULT_PULL_MESSAGE*

Ver también:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

Configures the language simplification policy, see *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configures the default *Control de acceso* for new projects, see *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configures the default value for *Acceso restringido* for new components, see *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configures the default value for *Permitir propagación de traducciones* for new components, see *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Configura *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Configura *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

Configura *DEFAULT_SHARED_TM*.

WEBLATE_AKISMET_API_KEY

Configura la clave de API de Akismet; vea *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Configura la firma con GPG de las consignas; vea *WEBLATE_GPG_IDENTITY*.

Ver también:

Signing Git commits with GnuPG

WEBLATE_URL_PREFIX

Configures URL prefix where Weblate is running, see *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Configures checks which you do not want to be displayed, see SILENCED_SYSTEM_CHECKS.

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Allows to customize `Content-Security-Policy` HTTP header.

Ver también:

Content security policy, *CSP_SCRIPT_SRC*, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

WEBLATE_LICENSE_FILTER

Configura *LICENSE_FILTER*.

WEBLATE_LICENSE_REQUIRED

Configura *LICENSE_REQUIRED*

WEBLATE_WEBSITE_REQUIRED

Configura *WEBSITE_REQUIRED*

WEBLATE_HIDE_VERSION

Configura *HIDE_VERSION*..

WEBLATE_BASIC_LANGUAGES

Configura *BASIC_LANGUAGES*.

WEBLATE_DEFAULT_AUTO_WATCH

Configura *DEFAULT_AUTO_WATCH*.

WEBLATE_RATELIMIT_ATTEMPTS

WEBLATE_RATELIMIT_LOCKOUT

WEBLATE_RATELIMIT_WINDOW

Nuevo en la versión 4.6.

Configura el limitador de velocidad.

Consejo: You can set configuration for any rate limiter scopes. To do that add `WEBLATE_` prefix to any of setting described in *Rate limiting*.

Ver también:

Rate limiting, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

WEBLATE_API_RATELIMIT_ANON

WEBLATE_API_RATELIMIT_USER

Nuevo en la versión 4.11.

Configures API rate limiting. Defaults to `100/day` for anonymous and `5000/hour` for authenticated users.

Ver también:

Limitación de la tasa de API

WEBLATE_ENABLE_HOOKS

Nuevo en la versión 4.13.

Configura *ENABLE_HOOKS*.

WEBLATE_ENABLE_AVATARS

Nuevo en la versión 4.6.1.

Configura *ENABLE_AVATARS*.

WEBLATE_AVATAR_URL_PREFIX

Nuevo en la versión 4.15.

Configura *AVATAR_URL_PREFIX*.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Nuevo en la versión 4.9.

Configura *LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH*.

WEBLATE_SSH_EXTRA_ARGS

Nuevo en la versión 4.9.

Configura *SSH_EXTRA_ARGS*.

WEBLATE_BORG_EXTRA_ARGS

Nuevo en la versión 4.9.

Configura *BORG_EXTRA_ARGS*.

WEBLATE_ENABLE_SHARING

Nuevo en la versión 4.14.1.

Configura *ENABLE_SHARING*.

WEBLATE_EXTRA_HTML_HEAD

Nuevo en la versión 4.15.

Configures *EXTRA_HTML_HEAD*.

WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE

Nuevo en la versión 4.15.

Configures *PRIVATE_COMMIT_EMAIL_TEMPLATE*.

WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN

Nuevo en la versión 4.15.

Configures *PRIVATE_COMMIT_EMAIL_OPT_IN*.

WEBLATE_UNUSED_ALERT_DAYS

Nuevo en la versión 4.17.

Configures *UNUSED_ALERT_DAYS*.

WEBLATE_CORS_ALLOWED_ORIGINS

Nuevo en la versión 4.16.

Allow CORS requests from given origins.

Ejemplo:

```
environment:
  WEBLATE_CORS_ALLOWED_ORIGINS: https://example.com,https://weblate.org
```

CLIENT_MAX_BODY_SIZE

Nuevo en la versión 4.16.3.

Configures maximal body size accepted by the built-in web server.

```
environment:
    CLIENT_MAX_BODY_SIZE: 200m
```

Consejo: This variable intentionally lacks `WEBLATE_` prefix as it is shared with third-party container used in *Certificados SSL automáticos con Let's Encrypt*.

Configuración de las sugerencias automáticas

Distinto en la versión 4.13: Los servicios de sugerencias automáticas ahora están configurados en la interfaz de usuario, consulte *Configuración de las sugerencias automáticas*.

The existing environment variables are imported during the migration to Weblate 4.13, but changing them will not have any further effect.

Configuración de autenticación

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

Ruta al archivo que contiene la contraseña de vinculación del servidor LDAP.

Ver también:

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

Configuración de la autenticación con LDAP.

Example for direct bind:

```
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
```

Example for search and bind:

```
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
```

Example for union search and bind:

```
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
```

Example with search and bind against Active Directory:

```
environment:
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS: 0
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER: (sAMAccountName=%(user)s)
```

Ver también:

Autenticación LDAP

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

Activa la *Autenticación por GitHub*.

GitHub Enterprise Edition

WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_URL

WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_API_URL

WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_SCOPE

Enables *GitHub EE authentication*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Activa la *Autenticación por Bitbucket*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Activa la *OAuth 2 de Facebook*.

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

Activa la *Google OAuth 2*.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

Activa la *OAuth 2 de GitLab*.

Gitea

WEBLATE_SOCIAL_AUTH_GITEA_API_URL

WEBLATE_SOCIAL_AUTH_GITEA_KEY

WEBLATE_SOCIAL_AUTH_GITEA_SECRET

Enables Gitea authentication.

Active Directory de Azure

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET

Enables Azure Active Directory authentication, see *Active Directory de Microsoft Azure*.

Azure Active Directory with Tenant support

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID

Enables Azure Active Directory authentication with Tenant support, see *Active Directory de Microsoft Azure*.

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

Enables Keycloak authentication, see documentation.

Proveedores de Linux

You can enable authentication using Linux vendors authentication services by setting following variables to any value.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_OPENINFRA

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Enables Slack authentication, see *Slack*.

OpenID Connect

Nuevo en la versión 4.13-1.

WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT

WEBLATE_SOCIAL_AUTH_OIDC_KEY

WEBLATE_SOCIAL_AUTH_OIDC_SECRET

WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY

Configures generic OpenID Connect integration.

Ver también:

OIDC (OpenID Connect)

SAML

Self-signed SAML keys are automatically generated on first container startup. In case you want to use own keys, place the certificate and private key in `/app/data/ssl/saml.crt` and `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

WEBLATE_SAML_IDP_IMAGE

WEBLATE_SAML_IDP_TITLE

SAML Identity Provider settings, see *Autenticación por SAML*.

WEBLATE_SAML_ID_ATTR_NAME

WEBLATE_SAML_ID_ATTR_USERNAME

WEBLATE_SAML_ID_ATTR_EMAIL

WEBLATE_SAML_ID_ATTR_USER_PERMANENT_ID

Nuevo en la versión 4.18.

SAML attributes mapping.

Otras configuraciones de autenticación

WEBLATE_NO_EMAIL_AUTH

Deshabilita la autenticación de correo electrónico cuando se establece en cualquier valor. Consulte *Desactivar la autenticación por contraseña*.

Puesta en marcha de la base de datos PostgreSQL

The database is created by `docker-compose.yml`, so these settings affect both Weblate and PostgreSQL containers.

Ver también:

Configuración de base de datos para Weblate

POSTGRES_PASSWORD

Contraseña de PostgreSQL.

POSTGRES_PASSWORD_FILE

Path to the file containing the PostgreSQL password. Use as an alternative to POSTGRES_PASSWORD.

POSTGRES_USER

Nombre de usuario de PostgreSQL.

POSTGRES_DATABASE

Nombre de base de datos de PostgreSQL.

POSTGRES_HOST

PostgreSQL server hostname or IP address. Defaults to `database`.

POSTGRES_PORT

PostgreSQL server port. Defaults to none (uses the default value).

POSTGRES_SSL_MODE

Configure how PostgreSQL handles SSL in connection to the server, for possible choices see SSL Mode Descriptions

POSTGRES_ALTER_ROLE

Configures name of role to alter during migrations, see *Configurar Weblate para que utilice PostgreSQL*.

POSTGRES_CONN_MAX_AGE

Nuevo en la versión 4.8.1.

The lifetime of a database connection, as an integer of seconds. Use 0 to close database connections at the end of each request (this is the default behavior).

Enabling connection persistence will typically, cause more open connection to the database. Please adjust your database configuration prior enabling.

Ejemplo de configuración:

```
environment:
    POSTGRES_CONN_MAX_AGE: 3600
```

Ver también:

CONN_MAX_AGE, Persistent connections

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

Nuevo en la versión 4.9.1.

Disable server side cursors in the database. This is necessary in some **pgbouncer** setups.

Ejemplo de configuración:

```
environment:
    POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Ver también:

DISABLE_SERVER_SIDE_CURSORS, Transaction pooling and server-side cursors

Configuración de copia de respaldo de la base de datos

Ver también:

Datos volcados para las copias de respaldo

WEBLATE_DATABASE_BACKUP

Configures the daily database dump using *DATABASE_BACKUP*. Defaults to `plain`.

Caching server setup

Using Redis is strongly recommended by Weblate and you have to provide a Redis instance when running Weblate in Docker.

Ver también:

Enable caching

REDIS_HOST

The Redis server hostname or IP address. Defaults to `cache`.

REDIS_PORT

The Redis server port. Defaults to `6379`.

REDIS_DB

The Redis database number, defaults to `1`.

REDIS_PASSWORD

La contraseña del servidor Redis, no utilizada de manera predeterminada.

REDIS_PASSWORD_FILE

Ruta al archivo que contiene la contraseña del servidor Redis.

Ver también:

REDIS_PASSWORD

REDIS_TLS

Permite el uso de SSL para la conexión con Redis.

REDIS_VERIFY_SSL

Se puede utilizar para desactivar la verificación de certificados SSL para la conexión con Redis.

Puesta en funcionamiento del servidor de correo

Para que funcione el correo saliente, debe proporcionar un servidor de correo.

Ejemplo de configuración de TLS:

```
environment:
    WEBLATE_EMAIL_HOST: smtp.example.com
    WEBLATE_EMAIL_HOST_USER: user
    WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Ejemplo de configuración de SSL:

```
environment:
    WEBLATE_EMAIL_HOST: smtp.example.com
    WEBLATE_EMAIL_PORT: 465
    WEBLATE_EMAIL_HOST_USER: user
    WEBLATE_EMAIL_HOST_PASSWORD: pass
    WEBLATE_EMAIL_USE_TLS: 0
    WEBLATE_EMAIL_USE_SSL: 1
```

Ver también:

Configurar el correo electrónico saliente

WEBLATE_EMAIL_HOST

Nombre de anfitrión o dirección IP del servidor de correo.

Ver también:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, *WEBLATE_EMAIL_USE_TLS*, EMAIL_HOST

WEBLATE_EMAIL_PORT

Puerto del servidor de correo, cuyo valor predeterminado es 25.

Ver también:

EMAIL_PORT

WEBLATE_EMAIL_HOST_USER

Usuario de autenticación del correo electrónico.

Ver también:

EMAIL_HOST_USER

WEBLATE_EMAIL_HOST_PASSWORD

Contraseña de autenticación del correo electrónico.

Ver también:

EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Ruta al archivo que contiene la contraseña de autenticación del correo electrónico.

Ver también:

WEBLATE_EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_USE_SSL

Whether to use an implicit TLS (secure) connection when talking to the SMTP server. In most e-mail documentation, this type of TLS connection is referred to as SSL. It is generally used on port 465. If you are experiencing problems, see the explicit TLS setting *WEBLATE_EMAIL_USE_TLS*.

Distinto en la versión 4.11: The SSL/TLS support is automatically enabled based on the *WEBLATE_EMAIL_PORT*.

Ver también:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_TLS*, EMAIL_USE_SSL

WEBLATE_EMAIL_USE_TLS

Whether to use a TLS (secure) connection when talking to the SMTP server. This is used for explicit TLS connections, generally on port 587 or 25. If you are experiencing connections that hang, see the implicit TLS setting *WEBLATE_EMAIL_USE_SSL*.

Distinto en la versión 4.11: The SSL/TLS support is automatically enabled based on the *WEBLATE_EMAIL_PORT*.

Ver también:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, EMAIL_USE_TLS

WEBLATE_EMAIL_BACKEND

Configura el dorsal Django para utilizarlo para enviar mensajes de correo electrónico.

Ver también:

Configure e-mail sending, EMAIL_BACKEND

WEBLATE_AUTO_UPDATE

Configura si Weblate debe actualizar los repositorios y cómo.

Ver también:

AUTO_UPDATE

Nota: This is a Boolean setting (use `"true"` or `"false"`).

Integración del sitio

WEBLATE_GET_HELP_URL

Configura *GET_HELP_URL*.

WEBLATE_STATUS_URL

Configura *STATUS_URL*.

WEBLATE_LEGAL_URL

Configura *LEGAL_URL*.

WEBLATE_PRIVACY_URL

Configura *PRIVACY_URL*.

Informe de defectos

Se recomienda recopilar sistemáticamente los errores que se producen en la instalación; vea ref:*collecting-errors*.

Para activar la compatibilidad con Rollbar, defina lo siguiente:

ROLLBAR_KEY

Su ficha de acceso POST al servidor de Rollbar.

ROLLBAR_ENVIRONMENT

Su entorno de Rollbar, cuyo valor predeterminado es `production`.

Para activar la compatibilidad con Sentry, defina lo siguiente:

SENTRY_DSN

Su DSN de Sentry.

SENTRY_ENVIRONMENT

Your Sentry Environment (optional), defaults to *WEBLATE_SITE_DOMAIN*.

SENTRY_TRACES_SAMPLE_RATE

Configure sampling rate for performance monitoring. Set to 1 to trace all events, 0 (the default) disables tracing.

Ejemplo:

```
environment:
  SENTRY_TRACES_SAMPLE_RATE: 0.5
```

SENTRY_PROFILES_SAMPLE_RATE

Configure sampling rate for profiling monitoring. Set to 1 to trace all events, 0 (the default) disables tracing.

Ejemplo:

```
environment:
  SENTRY_PROFILES_SAMPLE_RATE: 0.5
```

CDN de regionalización

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

Nuevo en la versión 4.2.1.

Configuración para *CDN de regionalización de JavaScript*.

The *WEBLATE_LOCALIZE_CDN_PATH* is path within the container. It should be stored on the persistent volume and not in the transient storage.

One of possibilities is storing that inside the Weblate data dir:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Nota: You are responsible for setting up serving of the files generated by Weblate, it only does stores the files in configured location.

Ver también:

weblate-cdn, *LOCALIZE_CDN_URL*, *LOCALIZE_CDN_PATH*

Cambiar las aplicaciones habilitadas, las comprobaciones, los complementos o las correcciones automáticas

Nuevo en la versión 3.8-5.

La configuración integrada de comprobaciones, complementos o correcciones automáticas habilitadas se puede ajustar mediante las siguientes variables:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Ejemplo:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
↪SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

Ver también:

CHECK_LIST, *AUTOFIX_LIST*, *WEBLATE_ADDONS*, INSTALLED_APPS

Configuración de contenedor

WEBLATE_WORKERS

Nuevo en la versión 4.6.1.

Base number of worker processes running in the container. When not set it is determined automatically on container startup based on number of CPU cores available.

It is used to determine *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS*, and *WEB_WORKERS*. You can use these settings to fine-tune.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

These variables allow you to adjust Celery worker options. It can be useful to adjust concurrency (`--concurrency 16`) or use different pool implementation (`--pool=gevent`).

By default, the number of concurrent workers is based on *WEBLATE_WORKERS*.

Ejemplo:

```
environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```

Ver también:

Celery worker options, *Tareas en segundo plano con Celery*

WEB_WORKERS

Configure how many uWSGI workers should be executed.

It defaults to *WEBLATE_WORKERS*.

Ejemplo:

```
environment:
  WEB_WORKERS: 32
```

WEBLATE_SERVICE

Defines which services should be executed inside the container. Use this for *Scaling horizontally*.

Se definen los siguientes servicios:

celery-beat
: Celery task scheduler, only one instance should be running. This container is also responsible for the database structure migrations and it should be started prior others.

celery-backup
: Celery worker for backups, only one instance should be running.

celery-celery
: Generic Celery worker.

celery-memory
: Memoria de traducción para programadores en Celery.

celery-notify
: Notificaciones para programadores de Celery.

celery-translate
: Traducción automática para programadored en Celey.

web
: Servidor web.

Volúmenes de contenedores Docker

There are two volumes (data and cache) exported by the Weblate container. The other service containers (PostgreSQL or Redis) have their data volumes as well, but those are not covered by this document.

The data volume is used to store Weblate persistent data such as cloned repositories or to customize Weblate installation.

The placement of the Docker volume on host system depends on your Docker configuration, but usually it is stored in `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/` (the path consist of name of your docker-compose directory, container, and volume names). In the container it is mounted as `/app/data`.

The cache volume is mounted as `/app/cache` and is used to store static files and `CACHE_DIR`. Its content is recreated on container startup and the volume can be mounted using ephemeral filesystem such as *tmpfs*.

When creating the volumes manually, the directories should be owned by UID 1000 as that is user used inside the container.

Ver también:

Docker volumes documentation

Read-only root filesystem

Nuevo en la versión 4.18.

When running the container with a read-only root filesytem, two additional *tmpfs* volumes are required - `/tmp` and `/run`.

Configuration beyond environment variables

Docker environment variables are intended to expose most *configuration settings* of relevance for Weblate installations.

If you find a setting that is not exposed as an environment variable, and you believe that it should be, feel free to *ask for it to be exposed in a future version of Weblate*.

If you need to modify a setting that is not exposed as a Docker environment variable, you can still do so, either *from the data volume* or *extending the Docker image*.

Ver también:

Personalizar Weblate

Overriding settings from the data volume

You can create a file at `/app/data/settings-override.py`, i.e. at the root of the *data volume*, to extend or override settings defined through environment variables.

Overriding settings by extending the Docker image

To override settings at the Docker image level instead of from the data volume:

1. *Create a custom Python package*.
2. Add a module to your package that imports all settings from `weblate.settings_docker`.

   For example, within the example package structure defined at *Crear un módulo Python*, you could create a file at `weblate_customization/weblate_customization/settings.py` with the following initial code:

```
from weblate.settings_docker import *
```

3. Create a custom `Dockerfile` that inherits from the official Weblate Docker image, and then installs your package and points the `DJANGO_SETTINGS_MODULE` environment variable to your settings module:

```
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. Instead of using the official Weblate Docker image, build a custom image from this `Dockerfile` file.

   There is no clean way to do this with `docker-compose.override.yml`. You *could* add `build: .` to the `weblate` node in that file, but then your custom image will be tagged as `weblate/weblate` in your system, which could be problematic.

   So, instead of using the `docker-compose.yml` straight from the official repository, unmodified, and extending it through `docker-compose.override.yml`, you may want to make a copy of the official `docker-compose.yml` file, and edit your copy to replace `image: weblate/weblate` with `build: .`.

   See the Compose file build reference for details on building images from source when using `docker-compose`.

5. Extend your custom settings module to define or redefine settings.

   You can define settings before or after the import statement above to determine which settings take precedence. Settings defined before the import statement can be overridden by environment variables and setting overrides defined in the data volume. Setting defined after the import statement cannot be overridden.

   You can also go further. For example, you can reproduce some of the things that `weblate.docker_settings` does, such as exposing settings as environment variables, or allow overriding settings from Python files in the data volume.

Replacing logo and other static files

Nuevo en la versión 3.8-5.

The static files coming with Weblate can be overridden by placing into `/app/data/python/customize/static` (see *Volúmenes de contenedores Docker*). For example creating `/app/data/python/customize/static/favicon.ico` will replace the favicon.

Consejo: The files are copied to the corresponding location upon container startup, so a restart of Weblate is needed after changing the content of the volume.

This approach can be also used to override Weblate templates. For example *Información legal* documents can be placed into `/app/data/python/customize/templates/legal/documents`.

Alternatively you can also include own module (see *Personalizar Weblate*) and add it as separate volume to the Docker container, for example:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/weblate_
```

(continué en la próxima página)

(proviene de la página anterior)

```
↪customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Configuración del servidor PostgreSQL

The PostgreSQL container uses default PostgreSQL configuration and it won't effectively utilize your CPU cores or memory. It is recommended to customize the configuration to improve the performance.

The configuration can be adjusted as described in *Database Configuration* at https://hub.docker.com/_/postgres. The configuration matching your environment can be generated using https://pgtune.leopard.in.ua/.

Container internals

The container is using **`supervisor`** to start individual services. In case of *Scaling horizontally*, it only starts single service in a container.

To check the services status use:

```
docker compose exec --user weblate weblate supervisorctl status
```

There are individual services for each Celery queue (see *Tareas en segundo plano con Celery* for details). You can stop processing some tasks by stopping the appropriate worker:

```
docker compose exec --user weblate weblate supervisorctl stop celery-translate
```

Instalar en Debian y Ubuntu

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espacio de almacenamiento

Cuanta más memoria tenga, mejor, ya que se utiliza para el prealmacenaje en todos los niveles (sistema de archivos, base de datos y Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

Nota: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

Instalación

Requisitos del sistema

Install the dependencies needed to build the Python modules (see *Requisitos de software*):

```
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
```

Instale las dependencias opcionales que desee en función de las funcionalidades que vaya a utilizar (vea *Dependencias opcionales*):

```
apt install -y \
   tesseract-ocr libtesseract-dev libleptonica-dev \
   libldap2-dev libldap-common libsasl2-dev \
   libxmlsec1-dev
```

Optionally install software for running production server, see *Running server*, *Configuración de base de datos para Weblate*, *Tareas en segundo plano con Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Las instrucciones de instalación local:

```
# 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
apt install -y redis-server

# Database server: PostgreSQL
apt install -y postgresql postgresql-contrib

# SMTP server
apt install -y exim4
```

Módulos de Python

Consejo: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Cree el entorno virtual para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Active el entorno virtual para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale Weblate incluyendo todas las dependencias opcionales:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

Please check *Dependencias opcionales* for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
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)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

---

1. Copy the file `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Adjusting configuration*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuración de base de datos para Weblate* for a production-ready setup):

   ```
   weblate migrate
   ```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

   ```
   weblate createadmin
   ```

5. Collect the static files for your web server (see *Running server* and *Serving static files*):

   ```
   weblate collectstatic
   ```

6. Compress the JavaScript and CSS files (optional, see *Compressing client assets*):

   ```
   weblate compress
   ```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tareas en segundo plano con Celery* has more info:

   ```
   ~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
   ```

8. Start the development server (*Running server* details a production setup):

   ```
   weblate runserver
   ```

After installation

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **`weblate check --deploy`**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

   The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* 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.

Instalar en SUSE y openSUSE

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espacio de almacenamiento

Cuanta más memoria tenga, mejor, ya que se utiliza para el prealmacenaje en todos los niveles (sistema de archivos, base de datos y Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

Nota: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

Instalación

Requisitos del sistema

Install the dependencies needed to build the Python modules (see *Requisitos de software*):

```
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
```

Instale las dependencias opcionales que desee en función de las funcionalidades que vaya a utilizar (vea *Dependencias opcionales*):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libsasl2-devel
zypper install libxmlsec1-devel
```

Optionally install software for running production server, see *Running server*, *Configuración de base de datos para Weblate*, *Tareas en segundo plano con Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Las instrucciones de instalación local:

```
# Web server option 1: NGINX and uWSGI
zypper install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
zypper install apache2 apache2-mod_wsgi

# Caching backend: Redis
zypper install redis-server

# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Módulos de Python

Consejo: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Cree el entorno virtual para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Active el entorno virtual para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale Weblate incluyendo todas las dependencias opcionales:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

Please check *Dependencias opcionales* for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
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)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

---

1. Copy the file `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Adjusting configuration*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuración de base de datos para Weblate* for a production-ready setup):

   ```
   weblate migrate
   ```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

   ```
   weblate createadmin
   ```

5. Collect the static files for your web server (see *Running server* and *Serving static files*):

   ```
   weblate collectstatic
   ```

6. Compress the JavaScript and CSS files (optional, see *Compressing client assets*):

   ```
   weblate compress
   ```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tareas en segundo plano con Celery* has more info:

   ```
   ~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
   ```

8. Start the development server (*Running server* details a production setup):

   ```
   weblate runserver
   ```

After installation

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **`weblate check --deploy`**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

   The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* 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.

Instalar en Red Hat, Fedora y CentOS

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espacio de almacenamiento

Cuanta más memoria tenga, mejor, ya que se utiliza para el prealmacenaje en todos los niveles (sistema de archivos, base de datos y Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

Nota: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

Instalación

Requisitos del sistema

Install the dependencies needed to build the Python modules (see *Requisitos de software*):

```
dnf install \
   libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
   libyaml-devel libffi-devel cairo-devel cairo-gobject-devel pango-devel \
   gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
   libtool-ltdl-devel python3-devel git
```

Instale las dependencias opcionales que desee en función de las funcionalidades que vaya a utilizar (vea *Dependencias opcionales*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
dnf install openldap-devel libsasl2-devel
dnf install xmlsec1-devel
```

Optionally install software for running production server, see *Running server*, *Configuración de base de datos para Weblate*, *Tareas en segundo plano con Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Las instrucciones de instalación local:

```
# Web server option 1: NGINX and uWSGI
dnf install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
dnf install apache2 apache2-mod_wsgi

# Caching backend: Redis
dnf install redis

# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Módulos de Python

Consejo: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Cree el entorno virtual para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Active el entorno virtual para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale Weblate incluyendo todas las dependencias opcionales:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

Please check *Dependencias opcionales* for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
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)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

---

1. Copy the file `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Adjusting configuration*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuración de base de datos para Weblate* for a production-ready setup):

   ```
   weblate migrate
   ```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

   ```
   weblate createadmin
   ```

5. Collect the static files for your web server (see *Running server* and *Serving static files*):

   ```
   weblate collectstatic
   ```

6. Compress the JavaScript and CSS files (optional, see *Compressing client assets*):

   ```
   weblate compress
   ```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tareas en segundo plano con Celery* has more info:

   ```
   ~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
   ```

8. Start the development server (*Running server* details a production setup):

   ```
   weblate runserver
   ```

After installation

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **`weblate check --deploy`**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

   The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* 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.

Instalar en macOS

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espacio de almacenamiento

Cuanta más memoria tenga, mejor, ya que se utiliza para el prealmacenaje en todos los niveles (sistema de archivos, base de datos y Weblate).

Many concurrent users increases the amount of needed CPU cores. For hundreds of translation components at least 4 GB of RAM is recommended.

The typical database storage usage is around 300 MB per 1 million hosted words. Storage space needed for cloned repositories varies, but Weblate tries to keep their size minimal by doing shallow clones.

Nota: Actual requirements for your installation of Weblate vary heavily based on the size of the translations managed in it.

Instalación

Requisitos del sistema

Install the dependencies needed to build the Python modules (see *Requisitos de software*):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip install virtualenv
```

Make sure pip will be able to find the `libffi` and `openssl` versions provided by homebrew — this will be needed during the installation build step.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig:/usr/local/opt/
↪openssl@3/lib/pkgconfig"
```

Instale las dependencias opcionales que desee en función de las funcionalidades que vaya a utilizar (vea *Dependencias opcionales*):

```
brew install tesseract
```

Optionally install software for running production server, see *Running server*, *Configuración de base de datos para Weblate*, *Tareas en segundo plano con Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Las instrucciones de instalación local:

```
# Web server option 1: NGINX and uWSGI
brew install nginx uwsgi

# Web server option 2: Apache with ``mod_wsgi``
brew install httpd

# Caching backend: Redis
brew install redis

# Database server: PostgreSQL
brew install postgresql
```

Módulos de Python

Consejo: We're using virtualenv to install Weblate in a separate environment from your system. If you are not familiar with it, check virtualenv User Guide.

1. Cree el entorno virtual para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Active el entorno virtual para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale Weblate incluyendo todas las dependencias opcionales:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Please check *Dependencias opcionales* for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
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)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Adjusting configuration*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuración de base de datos para Weblate* for a production-ready setup):

   ```
   weblate migrate
   ```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

   ```
   weblate createadmin
   ```

5. Collect the static files for your web server (see *Running server* and *Serving static files*):

   ```
   weblate collectstatic
   ```

6. Compress the JavaScript and CSS files (optional, see *Compressing client assets*):

   ```
   weblate compress
   ```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tareas en segundo plano con Celery* has more info:

   ```
   ~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
   ```

8. Start the development server (*Running server* details a production setup):

   ```
   weblate runserver
   ```

After installation

Congratulations, your Weblate server is now running and you can start using it.

- You can now access Weblate on `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **`weblate`** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **`weblate check --deploy`**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

   All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

   The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* 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.

Instalar desde el código fuente

1. Por favor, siga las instrucciones de la instalación de su sistema antes de instalar Weblate:

   - *Instalar en Debian y Ubuntu*
   - *Instalar en SUSE y openSUSE*
   - *Instalar en Red Hat, Fedora y CentOS*

2. Grab the latest Weblate sources using Git (or download a tarball and unpack that):

   ```
   git clone https://github.com/WeblateOrg/weblate.git weblate-src
   ```

   Alternatively you can use released archives. You can download them from our website <https://weblate.org/>. Those downloads are cryptographically signed, please see *Comprobar las firmas de versión*.

3. Install current Weblate code into the virtualenv:

   ```
   . ~/weblate-env/bin/activate
   pip install -e 'weblate-src[all]'
   ```

4. Copy `weblate/settings_example.py` to `weblate/settings.py`.
5. Adjust the values in the new `settings.py` file to your liking. You will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see *Adjusting configuration*.
6. Create the database used by Weblate, see *Configuración de base de datos para Weblate*.
7. Build Django tables, static files and initial data (see *Filling up the database* and *Serving static files*):

```
weblate migrate
weblate collectstatic
weblate compress
```

Nota: Este paso debe repetirse siempre que actualice el repositorio.

Instalar en OpenShift

With the OpenShift Weblate template you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database and persistent volume claims are used.

You can find the template at <https://github.com/WeblateOrg/openshift/>.

Instalación

Los ejemplos siguientes presuponen que cuenta con un entorno OpenShift versión 3.x en funcionamiento y la herramienta de cliente `oc` instalada. Consulte la documentación de OpenShift para obtener instrucciones.

The `template.yml` is suited for running all components in OpenShift. There is also `template-external-postgresql.yml` which does not start a PostgreSQL server and allows you to configure external PostgreSQL server.

Consola web

Copy the raw content from template.yml and import them into your project, then use the `Create` button in the OpenShift web console to create your application. The web console will prompt you for the values for all of the parameters used by the template.

CLI

To upload the Weblate template to your current project's template library, pass the `template.yml` file with the following command:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
   -n <PROJECT>
```

The template is now available for selection using the web console or the CLI.

Parámetros

The parameters that you can override are listed in the parameters section of the template. You can list them with the CLI by using the following command and specifying the file to be used:

```
$ 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
```

Aprovisionamiento

You can also use the CLI to process templates and use the configuration that is generated to create objects immediately.

```
$ 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
```

The Weblate instance should be available after successful migration and deployment at the specified `WEBLATE_SITE_DOMAIN` parameter.

After container setup, you can sign in as *admin* user with password provided in `WEBLATE_ADMIN_PASSWORD`, or a random password generated on first start if that was not set.

To reset *admin* password, restart the container with `WEBLATE_ADMIN_PASSWORD` set to new password in the respective `Secret`.

Eliminar

```
$ oc delete all -l app=<APPLICATION_NAME>
$ oc delete configmap -l app= <APPLICATION_NAME>
$ oc delete secret -l app=<APPLICATION_NAME>
# ATTTENTION! 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
```

Configuración

By processing the template a respective `ConfigMap` will be created and which can be used to customize the Weblate image. The `ConfigMap` is directly mounted as environment variables and triggers a new deployment every time it is changed. For further configuration options, see *Docker environment variables* for full list of environment variables.

Instalación en Kubernetes

Nota: This guide is looking for contributors experienced with Kubernetes to cover the setup in more details.

With the Kubernetes Helm chart you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database and persistent volume claims are used.

You can find the chart at <https://github.com/WeblateOrg/helm/> and it can be displayed at <https://artifacthub.io/packages/helm/weblate/weblate>.

Instalación

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Configuración

For further configuration options, see *Docker environment variables* for full list of environment variables.

En función de la preparación y su experiencia, elija un método de instalación apropiado para usted:

- *Instalar con Docker*, recomendable para montajes en entornos de producción.
- Instalación en entorno virtual, recomendable para montajes en entornos de producción:
  - *Instalar en Debian y Ubuntu*
  - *Instalar en SUSE y openSUSE*
  - *Instalar en Red Hat, Fedora y CentOS*
  - *Instalar en macOS*
- *Instalar desde el código fuente*, recomendable para el desarrollo.
- *Instalar en OpenShift*
- *Instalación en Kubernetes*

2.1.2 Requisitos de software

Sistema operativo

Se sabe que Weblate funciona en Linux, FreeBSD y macOS. Es posible que funcione también en otros sistemas similares a Unix.

Weblate no es compatible con Windows. Aun así, es posible hacerlo funcionar; aceptaremos parches para este fin.

Otros servicios

Weblate utiliza otros servicios para su funcionamiento. Habrá de ejecutar al menos los siguientes:

- PostgreSQL database server, see *Configuración de base de datos para Weblate*.
- Redis server for cache and tasks queue, see *Tareas en segundo plano con Celery*.
- SMTP server for outgoing e-mail, see *Configurar el correo electrónico saliente*.

Dependencias de Python

Weblate is written in Python and supports Python 3.6 or newer. You can install dependencies using pip or from your distribution packages, full list is available in `requirements.txt`.

Dependencias más notables:

Django
https://www.djangoproject.com/

Celery
https://docs.celeryq.dev/

Translate Toolkit
https://toolkit.translatehouse.org/

buscador de traducciones
https://github.com/WeblateOrg/translation-finder

Python Social Auth
https://python-social-auth.readthedocs.io/

Marco REST de Django
https://www.django-rest-framework.org/

Dependencias opcionales

Los módulos siguientes son necesarios para algunas funcionalidades de Weblate: Los encontrará todos en `requirements-optional.txt`.

`Mercurial` (opcional para soporte de repositorios *Mercurial*)
https://www.mercurial-scm.org/

`phply` (opcional para *Cadenas de PHP*)
https://github.com/viraptor/phply

`tesserocr` (opcional para OCR en *Contexto visual para cadenas*)
https://github.com/sirfz/tesserocr

`python-akismet` (opcional para *Spam protection*)
https://github.com/Nekmo/python-akismet

`ruamel.yaml` (opcional para *YAML files*)
https://pypi.org/project/ruamel.yaml/

`Zeep` (opcional para *Microsoft Terminology*)
https://docs.python-zeep.org/

`aeidon` (opcional para *Archivos de subtítulos*)
https://pypi.org/project/aeidon/

`fluent.syntax` (opcional para *Fluent format*)
https://projectfluent.org/

Consejo: When installing using pip, you can directly specify desired features when installing:

```
pip install "Weblate[PHP,Fluent]"
```

O puede instalar Weblate con todas las funciones opcionales:

```
pip install "Weblate[all]"
```

Or you can install Weblate without any optional features:

```
pip install Weblate
```

Dependencias del motor de la base de datos

Weblate supports PostgreSQL, MySQL and MariaDB, see *Configuración de base de datos para Weblate* and backends documentation for more details.

Otros requisitos de sistema

Deben instalarse las dependencias siguientes en el sistema:

`Git`
https://git-scm.com/

Pango, Cairo and related header files and GObject introspection data
https://cairographics.org/, https://pango.gnome.org/, see *Pango y Cairo*

`git-review` (opcional para admitir Gerrit)
https://pypi.org/project/git-review/

`git-svn` (opcional para admitir Subversion)
https://git-scm.com/docs/git-svn

`tesseract` y sus datos (opcional para el reconocimiento óptico de caracteres en capturas de pantalla)
https://github.com/tesseract-ocr/tesseract

`licensee` (optional for detecting license when creating component)
https://github.com/licensee/licensee

Dependencias en tiempo de la compilación

To build some of the *Dependencias de Python* you might need to install their dependencies. This depends on how you install them, so please consult individual packages for documentation. You won't need those if using prebuilt `Wheels` while installing using `pip` or when you use distribution packages.

Pango y Cairo

Distinto en la versión 3.7.

Weblate uses Pango and Cairo for rendering bitmap widgets (see promotion) and rendering checks (see *Gestionar tipos de letra*). To properly install Python bindings for those you need to install system libraries first - you need both Cairo and Pango, which in turn need GLib. All those should be installed with development files and GObject introspection data.

2.1.3 Comprobar las firmas de versión

Weblate release are cryptographically signed by the releasing developer. Currently this is Michal Čihař. Fingerprint of his PGP key is:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

and you can get more identification information from <https://keybase.io/nijel>.

Debería verificar que coincida la firma con el archivador que ha descargado. Así, podrá tener la certeza de que está utilizando el mismo código que fue publicado. Además, compruebe la fecha de la firma para asegurarse de que ha descargado la versión más reciente.

Todos los archivadores incluyen un archivo `.asc`, que contiene la firma PGP correspondiente. Coloque el archivador y el archivo de firma en la misma carpeta para verificarlos:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

Como podrá darse cuenta, GPG responde que no se conoce la clave pública. Efectúe uno de los procedimientos siguientes:

- Utilice *wkd* para descargar la clave:

```
$ 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 (Braiins) <michal.cihar@braiins.cz>
sub   rsa4096 2009-06-17 [E]
sub   rsa4096 2015-09-09 [S]
```

- Descargue el llavero del servidor de Michal y, a continuación, impórtelo con:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Descargue e importe la clave de uno de los servidores de claves:

```
$ 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
```

This will improve the situation a bit - at this point you can verify that the signature from the given key is correct but you still can not trust the name used in the key:

```
$ 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 (Braiins) <michal.cihar@braiins.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
```

The problem here is that anybody could issue the key with this name. You need to ensure that the key is actually owned by the mentioned person. The GNU Privacy Handbook covers this topic in the chapter Validating other keys on your public keyring. The most reliable method is to meet the developer in person and exchange key fingerprints, however you can also rely on the web of trust. This way you can trust the key transitively through signatures of others, who have met the developer in person.

Cuando la clave sea de fiar, el aviso dejará de emitirse:

```
$ 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]
```

(continué en la próxima página)

(proviene de la página anterior)

```
gpg:                 aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:                 aka "[jpeg image of size 8848]" [ultimate]
gpg:                 aka "Michal Čihař (Braiins) <michal.cihar@braiins.cz>"␣
↪[ultimate]
```

Should the signature be invalid (the archive has been changed), you would get a clear error regardless of the fact that the key is trusted or not:

```
$ 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 Permisos del sistema de archivos

The Weblate process needs to be able to read and write to the directory where it keeps data - *DATA_DIR*. All files within this directory should be owned and writable by the user running all Weblate processes (typically WSGI and Celery, see *Running server* and *Tareas en segundo plano con Celery*).

The default configuration places them in the same tree as the Weblate sources, however you might prefer to move these to a better location such as: `/var/lib/weblate`.

Weblate tries to create these directories automatically, but it will fail when it does not have permissions to do so.

You should also take care when running *Órdenes de gestión*, as they should be ran under the same user as Weblate itself is running, otherwise permissions on some files might be wrong.

In the Docker container, all files in the `/app/data` volume have to be owned by the `weblate` user inside the container (UID 1000).

Ver también:

Serving static files

2.1.5 Configuración de base de datos para Weblate

Es recomendable ejecutar Weblate con un servidor de bases de datos PostgreSQL.

PostgreSQL 12 and higher is supported.

Ver también:

Use a powerful database engine, Databases, *Migrating from other databases to PostgreSQL*

PostgreSQL

PostgreSQL is usually the best choice for Django-based sites. It's the reference database used for implementing Django database layer.

Nota: Weblate uses trigram extension which has to be installed separately in some cases. Look for `postgresql-contrib` or a similarly named package.

Ver también:

PostgreSQL notes

Crear una base de datos en PostgreSQL

Suele ser una buena idea ejecutar Weblate en su propia base de datos, en una cuenta de usuario separada:

```
# 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
```

Consejo: If you don't want to make the Weblate user a superuser in PostgreSQL, you can omit that. In that case you will have to perform some of the migration steps manually as a PostgreSQL superuser in schema Weblate will use:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm;
CREATE EXTENSION IF NOT EXISTS btree_gin;
```

Configurar Weblate para que utilice PostgreSQL

The `settings.py` snippet for PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

The database migration performs ALTER ROLE on database role used by Weblate. In most cases the name of the role matches username. In more complex setups the role name is different than username and you will get error about non-existing role during the database migration (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). This is known to happen with Azure Database for PostgreSQL, but it's not limited to this environment. Please set `ALTER_ROLE` to change name of the role Weblate should alter during the database migration.

MySQL y MariaDB

Advertencia: While MySQL and MariaDB support is still maintained in Weblate, our primary focus is PostgreSQL. It is recommended to use PostgreSQL for new installs, and to migrate existing installs to PostgreSQL, see *Migrating from other databases to PostgreSQL*.

Some Weblate features will perform better with *PostgreSQL*. This includes searching and translation memory, which both utilize full-text features in the database and PostgreSQL implementation is superior.

Weblate can be also used with MySQL or MariaDB, please see MySQL notes and MariaDB notes for caveats using Django with those. Because of the limitations it is recommended to use *PostgreSQL* for new installations.

Weblate requires MySQL at least 8 or MariaDB at least 10.4.

Se recomienda la configuración siguiente para Weblate:

- Utilice el conjunto de caracteres `utf8mb4` para permitir la representación de los planos superiores de Unicode (donde se encuentran, por ejemplo, los emoyis).
- Configure the server with `innodb_large_prefix` to allow longer indices on text fields.
- Set the isolation level to `READ COMMITTED`.
- The SQL mode should be set to `STRICT_TRANS_TABLES`.

MySQL 8.x, MariaDB 10.5.x or newer have reasonable default configuration so that no server tweaking should be necessary and all what is needed can be configured on the client side.

El siguiente es un **:archivo:`/etc/my.cnf.d/server.cnf`** de ejemplo para un servidor con 8 GB de RAM. Esta configuración debería bastar en la mayoría de las instalaciones. MySQL y MariaDB poseen configuraciones que optimizarán el rendimiento de su servidor, las cuales no es necesario ajustar a menos que espere tener cantidades elevadas de usuarios accediendo al sistema en simultáneo. Consulte la documentación del proveedor de su base de datos para obtener más información al respecto.

It is absolutely critical to reduce issues when installing that the setting `innodb_file_per_table` is set properly and MySQL/MariaDB restarted before you start your Weblate install.

```
[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
```

Consejo: In case you are getting `#1071 - Specified key was too long; max key length is 767 bytes` error, please update your configuration to include the `innodb` settings above and restart your install.

Consejo: In case you are getting `#2006 - MySQL server has gone away` error, configuring `CONN_MAX_AGE` might help.

Configuración de Weblate para utilizar MySQL/MariaDB

The `settings.py` snippet for MySQL and MariaDB:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "127.0.0.1",
        # Set to empty string for default
        "PORT": "3306",
        # In case you wish to use additional
        # connection options
        "OPTIONS": {},
    }
}
```

You should also create the `weblate` user account in MySQL or MariaDB before you begin the install. Use the commands below to achieve that:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Otras configuraciones

Configurar el correo electrónico saliente

Weblate sends out e-mails on various occasions - for account activation and on various notifications configured by users. For this it needs access to an SMTP server.

The mail server setup is configured using these settings: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_TLS`, `EMAIL_USE_SSL`, `EMAIL_HOST_USER` and `EMAIL_PORT`. Their names are quite self-explanatory, but you can find more info in the Django documentation.

Consejo: In case you get error about not supported authentication (for example `SMTP AUTH extension not supported by server`), it is most likely caused by using insecure connection and server refuses to authenticate this way. Try enabling `EMAIL_USE_TLS` in such case.

Ver también:

Not receiving e-mails from Weblate, *Configuring outgoing e-mail in Docker container*

Running behind reverse proxy

Several features in Weblate rely on being able to get client IP address. This includes *Rate limiting*, *Spam protection* or *Registro de auditoría*.

In default configuration Weblate parses IP address from `REMOTE_ADDR` which is set by the WSGI handler.

In case you are running a reverse proxy, this field will most likely contain its address. You need to configure Weblate to trust additional HTTP headers and parse the IP address from these. This can not be enabled by default as it would allow IP address spoofing for installations not using a reverse proxy. Enabling *IP_BEHIND_REVERSE_PROXY* might be enough for the most usual setups, but you might need to adjust *IP_PROXY_HEADER* and *IP_PROXY_OFFSET* as well.

Another thing to take care of is the Host header. It should match to whatever is configured as *SITE_DOMAIN*. Additional configuration might be needed in your reverse proxy (for example use `ProxyPreserveHost On` for Apache or `proxy_set_header Host $host;` with nginx).

Ver también:

Spam protection, *Rate limiting*, *Registro de auditoría*, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_HEADER*, *IP_PROXY_OFFSET*, SECURE_PROXY_SSL_HEADER

HTTP proxy

Weblate does execute VCS commands and those accept proxy configuration from environment. The recommended approach is to define proxy settings in `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Ver también:

Proxy Environment Variables

2.1.7 Adjusting configuration

Ver también:

Configuración de muestra

Copy `weblate/settings_example.py` to `weblate/settings.py` and adjust it to match your setup. You will probably want to adjust the following options: `ADMINS`

List of site administrators to receive notifications when something goes wrong, for example notifications on failed merges, or Django errors.

Ver también:

ADMINS, *Properly configure admins*

`ALLOWED_HOSTS`

You need to set this to list the hosts your site is supposed to serve. For example:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternatively you can include wildcard:

```
ALLOWED_HOSTS = ["*"]
```

Ver también:

ALLOWED_HOSTS, *WEBLATE_ALLOWED_HOSTS*, *Allowed hosts setup*

SESSION_ENGINE

Configure how your sessions will be stored. In case you keep the default database backend engine, you should schedule: **`weblate clearsessions`** to remove stale session data from the database.

If you are using Redis as cache (see *Enable caching*) it is recommended to use it for sessions as well:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Ver también:

Configuring the session engine, SESSION_ENGINE

DATABASES

Connectivity to database server, please check Django's documentation for more details.

Ver también:

Configuración de base de datos para Weblate, DATABASES, Databases

DEBUG

Disable this for any production server. With debug mode enabled, Django will show backtraces in case of error to users, when you disable it, errors will be sent per e-mail to ADMINS (see above).

Debug mode also slows down Weblate, as Django stores much more info internally in this case.

Ver también:

DEBUG, *Desactivar el modo de depuración*

DEFAULT_FROM_EMAIL

E-mail sender address for outgoing e-mail, for example registration e-mails.

Ver también:

DEFAULT_FROM_EMAIL

SECRET_KEY

Key used by Django to sign some info in cookies, see *Clave secreta de Django* for more info.

Ver también:

SECRET_KEY

SERVER_EMAIL

E-mail used as sender address for sending e-mails to the administrator, for example notifications on failed merges.

Ver también:

SERVER_EMAIL

2.1.8 Filling up the database

After your configuration is ready, you can run *`weblate weblate migrate`* to create the database structure. Now you should be able to create translation projects using the admin interface.

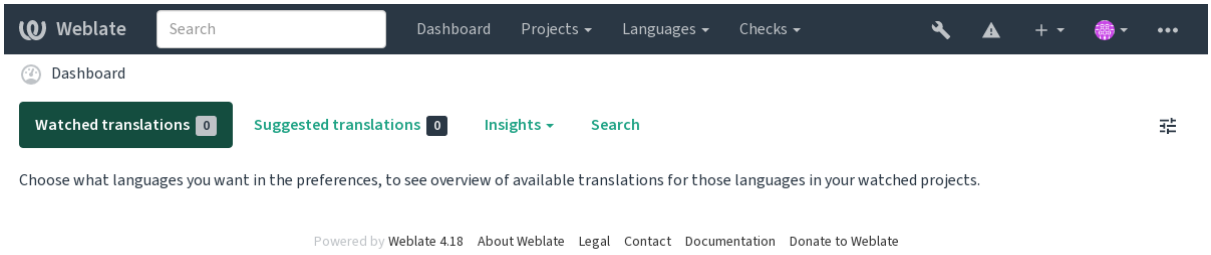
Once you are done, you should also check the *Performance report* in the admin interface, which will give you hints of potential non optimal configuration on your site.

Ver también:

Configuración, *Lista de privilegios y roles incorporados*

2.1.9 Puesta en marcha de entorno de producción

For a production setup you should carry out adjustments described in the following sections. The most critical settings will trigger a warning, which is indicated by an exclamation mark in the top bar if signed in as a superuser:



It is also recommended to inspect checks triggered by Django (though you might not need to fix all of them):

```
weblate check --deploy
```

You can also review the very same checklist from the *Interfaz de gestión*.

Ver también:

Deployment checklist

Desactivar el modo de depuración

Ejecute esto para desactivar el modo de depuración (`DEBUG`) de Django:

```
DEBUG = False
```

Con el modo de depuración activado, Django almacena todas las consultas ejecutadas y muestra a los usuarios el seguimiento regresivo de los errores, lo cual no es deseable en un entorno de producción.

Ver también:

Adjusting configuration

Properly configure admins

Set the correct admin addresses to the `ADMINS` setting to defining who will receive e-mails in case something goes wrong on the server, for example:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Ver también:

Adjusting configuration

Set correct site domain

Adjust site name and domain in the admin interface, otherwise links in RSS or registration e-mails will not work. This is configured using `SITE_DOMAIN` which should contain site domain name.

Distinto en la versión 4.2: Prior to the 4.2 release the Django sites framework was used instead, please see The «sites» framework.

Ver también:

Allowed hosts setup, *Correctly configure HTTPS* `SITE_DOMAIN`, `WEBLATE_SITE_DOMAIN`, `ENABLE_HTTPS`

Correctly configure HTTPS

It is strongly recommended to run Weblate using the encrypted HTTPS protocol. After enabling it, you should set *ENABLE_HTTPS* in the settings:

```
ENABLE_HTTPS = True
```

Consejo: You might want to set up HSTS as well, see SSL/HTTPS for more details.

Ver también:

ENABLE_HTTPS, *Allowed hosts setup*, *Set correct site domain*

Set properly SECURE_HSTS_SECONDS

If your site is served over SSL, you have to consider setting a value for `SECURE_HSTS_SECONDS` in the `settings.py` to enable HTTP Strict Transport Security. By default it's set to 0 as shown below.

```
SECURE_HSTS_SECONDS = 0
```

If set to a non-zero integer value, the `django.middleware.security.SecurityMiddleware` sets the HTTP Strict Transport Security header on all responses that do not already have it.

Advertencia: Setting this incorrectly can irreversibly (for some time) break your site. Read the HTTP Strict Transport Security documentation first.

Use a powerful database engine

- Please use PostgreSQL for a production environment, see *Configuración de base de datos para Weblate* for more info.
- Use adjacent location for running the database server, otherwise the networking performance or reliability might ruin your Weblate experience.
- Check the database server performance or tweak its configuration, for example using PGTune.

Ver también:

Configuración de base de datos para Weblate, *Migrating from other databases to PostgreSQL*, *Adjusting configuration*, Databases

Enable caching

If possible, use Redis from Django by adjusting the `CACHES` configuration variable, for example:

```
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",
        },
```

(continué en la próxima página)

(proviene de la página anterior)

```
    }
}
```

Consejo: In case you change Redis settings for the cache, you might need to adjust them for Celery as well, see *Tareas en segundo plano con Celery*.

Ver también:

Caché de avatars, Django's cache framework

Caché de avatars

In addition to caching of Django, Weblate performs caching of avatars. It is recommended to use a separate, file-backed cache for this purpose:

```
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,
        },
    },
}
```

Ver también:

ENABLE_AVATARS, *AVATAR_URL_PREFIX*, *Avatars*, *Enable caching*, Django's cache framework

Configure e-mail sending

Weblate needs to send out e-mails on several occasions, and these e-mails should have a correct sender address, please configure *SERVER_EMAIL* and *DEFAULT_FROM_EMAIL* to match your environment, for example:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Nota: To disable sending e-mails by Weblate set EMAIL_BACKEND to `django.core.mail.backends.dummy.EmailBackend`.

This will disable *all* e-mail delivery including registration or password reset e-mails.

Ver también:

Adjusting configuration, *Configurar el correo electrónico saliente*, EMAIL_BACKEND, DEFAULT_FROM_EMAIL, SERVER_EMAIL

Allowed hosts setup

Django requires *ALLOWED_HOSTS* to hold a list of domain names your site is allowed to serve, leaving it empty will block any requests.

In case this is not configured to match your HTTP server, you will get errors like `Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.`

Consejo: On Docker container, this is available as *WEBLATE_ALLOWED_HOSTS*.

Ver también:

ALLOWED_HOSTS, *WEBLATE_ALLOWED_HOSTS*, *Set correct site domain*

Clave secreta de Django

The *SECRET_KEY* setting is used by Django to sign cookies, and you should really generate your own value rather than using the one from the example setup.

You can generate a new key using **`weblate-generate-secret-key`** shipped with Weblate.

Ver también:

SECRET_KEY

Efectuar tareas de mantenimiento

For optimal performance, it is good idea to run some maintenance tasks in the background. This is now automatically done by *Tareas en segundo plano con Celery* and covers following tasks:

- Configuration health check (hourly).
- Committing pending changes (hourly), see *Consignas diferidas* and *weblate weblate commit_pending*.
- Updating component alerts (daily).
- Update remote branches (nightly), see *AUTO_UPDATE*.
- Translation memory backup to JSON (daily), see *weblate weblate dump_memory*.
- Fulltext and database maintenance tasks (daily and weekly tasks), see *weblate weblate cleanup-trans*.

Distinto en la versión 3.2: Since version 3.2, the default way of executing these tasks is using Celery and Weblate already comes with proper configuration, see *Tareas en segundo plano con Celery*.

System locales and encoding

The system locales should be configured to UTF-8 capable ones. On most Linux distributions this is the default setting. In case it is not the case on your system, please change locales to UTF-8 variant.

For example by editing `/etc/default/locale` and setting there `LANG="C.UTF-8"`.

In some cases the individual services have separate configuration for locales. This varies between distribution and web servers, so check documentation of your web server packages for that.

Apache on Ubuntu uses `/etc/apache2/envvars`:

```
export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'
```

Apache on CentOS uses `/etc/sysconfig/httpd` (or `/opt/rh/httpd24/root/etc/sysconfig/httpd`):

```
LANG='en_US.UTF-8'
```

Using custom certificate authority

Weblate does verify SSL certificates during HTTP requests. In case you are using custom certificate authority which is not trusted in default bundles, you will have to add its certificate as trusted.

The preferred approach is to do this at system level, please check your distro documentation for more details (for example on debian this can be done by placing the CA certificate into `/usr/local/share/ca-certificates/` and running **`update-ca-certificates`**).

Once this is done, system tools will trust the certificate and this includes Git.

For Python code, you will need to configure requests to use system CA bundle instead of the one shipped with it. This can be achieved by placing following snippet to `settings.py` (the path is Debian specific):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Compressing client assets

Weblate comes with a bunch of JavaScript and CSS files. For performance reasons it is good to compress them before sending to a client. In default configuration this is done on the fly at cost of little overhead. On big installations, it is recommended to enable offline compression mode. This needs to be done in the configuration and the compression has to be triggered on every Weblate upgrade.

The configuration switch is simple by enabling `django.conf.settings.COMPRESS_OFFLINE` and configuring `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (the latter is already included in the example configuration):

```
COMPRESS_OFFLINE = True
```

On each deploy you need to compress the files to match current version:

```
weblate compress
```

Consejo: La imagen oficial para Docker ya tiene activada esta funcionalidad.

Ver también:

Common Deployment Scenarios, *Serving static files*

2.1.10 Running server

Consejo: In case you are not experienced with services described below, you might want to try *Instalar con Docker*.

Es necesario contar con varios servicios para ejecutar Weblate. El montaje recomendado consiste de:

- Servidor de base de datos (vea *Configuración de base de datos para Weblate*)
- Servidor de antememoria (vea *Enable caching*)
- Frontend web server for static files and SSL termination (see *Serving static files*)

- Servidor WSGI para el contenido dinámico (vea *Sample configuration for NGINX and uWSGI*)
- Celery para ejecutar las tareas en segundo plano (vea *Tareas en segundo plano con Celery*)

Nota: There are some dependencies between the services, for example cache and database should be running when starting up Celery or uwsgi processes.

In most cases, you will run all services on single (virtual) server, but in case your installation is heavy loaded, you can split up the services. The only limitation on this is that Celery and Wsgi servers need access to *DATA_DIR*.

Nota: The WSGI process has to be executed under the same user the Celery process, otherwise files in the *DATA_DIR* will be stored with mixed ownership, leading to runtime issues.

See also *Permisos del sistema de archivos* and *Tareas en segundo plano con Celery*.

Running web server

Running Weblate is not different from running any other Django based program. Django is usually executed as uWSGI or fcgi (see examples for different webservers below).

For testing purposes, you can use the built-in web server in Django:

```
weblate runserver
```

Advertencia: DO NOT USE THIS SERVER IN A PRODUCTION SETTING. It has not gone through security audits or performance tests. See also Django documentation on `runserver`.

Consejo: The Django built-in server serves static files only with *DEBUG* enabled as it is intended for development only. For production use, please see wsgi setups in *Sample configuration for NGINX and uWSGI*, *Sample configuration for Apache*, *Sample configuration for Apache and Gunicorn*, and *Serving static files*.

Serving static files

Django needs to collect its static files in a single directory. To do so, execute `weblate collectstatic --noinput`. This will copy the static files into a directory specified by the `STATIC_ROOT` setting (this defaults to a `static` directory inside *DATA_DIR*).

It is recommended to serve static files directly from your web server, you should use that for the following paths:

`/static/`
: Serves static files for Weblate and the admin interface (from defined by `STATIC_ROOT`).

`/media/`
: Used for user media uploads (e.g. screenshots).

`/favicon.ico`
: Should be rewritten to rewrite a rule to serve `/static/favicon.ico`.

Ver también:

Sample configuration for NGINX and uWSGI, *Sample configuration for Apache*, *Sample configuration for Apache and Gunicorn*, *Compressing client assets*, How to deploy Django, How to deploy static files

Content security policy

The default Weblate configuration enables `weblate.middleware.SecurityMiddleware` middleware which sets security related HTTP headers like `Content-Security-Policy` or `X-XSS-Protection`. These are by default set up to work with Weblate and its configuration, but this might need customization for your environment.

Ver también:

CSP_SCRIPT_SRC, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

Sample configuration for NGINX and uWSGI

To run production webserver, use the wsgi wrapper installed with Weblate (in virtual env case it is installed as `~/weblate-env/lib/python3.9/site-packages/weblate/wsgi.py`). Don't forget to set the Python search path to your virtualenv as well (for example using `virtualenv = /home/user/weblate-env` in uWSGI).

The following configuration runs Weblate as uWSGI under the NGINX webserver.

Configuration for NGINX (also available as `weblate/examples/weblate.nginx.conf`):

```
#
# nginx configuration for Weblate
#
# You will want to change:
#
# - server_name
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - 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:
```

(continué en la próxima página)

(proviene de la página anterior)

```
        uwsgi_pass unix:///run/uwsgi/app/weblate/socket;
        # uwsgi_pass 127.0.0.1:8080;
    }
}
```

Configuration for uWSGI (also available as `weblate/examples/weblate.uwsgi.ini`):

```
#
# uWSGI configuration for Weblate
#
# You will want to change:
#
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
[uwsgi]
plugins       = python3
master        = true
protocol      = uwsgi
socket        = 127.0.0.1:8080
wsgi-file     = /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/wsgi.
↪py

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path   = /path/to/weblate

# 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
```

(continué en la próxima página)

(proviene de la página anterior)

```
# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

Ver también:

How to use Django with uWSGI

Sample configuration for Apache

It is recommended to use prefork MPM when using WSGI with Weblate.

The following configuration runs Weblate as WSGI, you need to have enabled `mod_wsgi` (available as `weblate/examples/apache.conf`):

```
#
# VirtualHost for Weblate
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match Python version mod-wsgi is compiled for
# - 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
        </Files>
```

(continué en la próxima página)

(proviene de la página anterior)

```
    </Directory>

</VirtualHost>
```

Nota: Weblate requires Python 3, so please make sure you are running Python 3 variant of the modwsgi. Usually it is available as a separate package, for example `libapache2-mod-wsgi-py3`.

Use matching Python version to install Weblate.

Ver también:

System locales and encoding, Como usar Django con Apache y mod_wsgi

Sample configuration for Apache and Gunicorn

The following configuration runs Weblate in Gunicorn and Apache 2.4 (available as `weblate/examples/apache.gunicorn.conf`):

```
#
# VirtualHost for Weblate using gunicorn on localhost:8000
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - 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/
```

(continué en la próxima página)

(proviene de la página anterior)

```
    ProxyPreserveHost On
</VirtualHost>
```

Ver también:

How to use Django with Gunicorn

Running Weblate under path

It is recommended to use prefork MPM when using WSGI with Weblate.

A sample Apache configuration to serve Weblate under `/weblate`. Again using `mod_wsgi` (also available as `weblate/examples/apache-path.conf`):

```
#
# VirtualHost for Weblate, running under /weblate path
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match Python version mod-wsgi is compiled for
# - 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>

</VirtualHost>
```

Additionally, you will have to adjust `weblate/settings.py`:

```
URL_PREFIX = "/weblate"
```

2.1.11 Tareas en segundo plano con Celery

Nuevo en la versión 3.2.

Weblate uses Celery to execute regular and background tasks. You are supposed to run a Celery service that will execute these. For example, it is responsible for handling following operations (this list is not complete):

- Receiving webhooks from external services (see *Actuadores de notificación*).
- Running regular maintenance tasks such as backups, cleanups, daily add-ons, or updates (see *Respaldar y trasladar Weblate*, *BACKGROUND_TASKS*, *Complementos*).
- Ejecutando *Traducción automática*.
- Enviar notificaciones con el resumen.
- Offloading expensive operations from the wsgi process.
- Confirmar los cambios pendientes (ver *Consignas diferidas*).

A typical setup using Redis as a backend looks like this:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Ver también:

Redis broker configuration in Celery

You should also start the Celery worker to process the tasks and start scheduled tasks, this can be done directly on the command-line (which is mostly useful when debugging or developing):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Nota: The Celery process has to be executed under the same user as the WSGI process, otherwise files in the *DATA_DIR* will be stored with mixed ownership, leading to runtime issues.

See also *Permisos del sistema de archivos* and *Running server*.

Executing Celery tasks in the wsgi using eager mode

Nota: This will have severe performance impact on the web interface, and will break features depending on regular trigger (for example committing pending changes, digest notifications, or backups).

For development, you might want to use eager configuration, which does process all tasks in place:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Running Celery as system service

Most likely you will want to run Celery as a daemon and that is covered by Daemonization. For the most common Linux setup using systemd, you can use the example files shipped in the `examples` folder listed below.

Systemd unit to be placed as `/etc/systemd/system/celery-weblate.service`:

```
[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
```

Environment configuration to be placed as `/etc/default/celery-weblate`:

```
# 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"
```

Additional configuration to rotate Celery logs using **`logrotate`** to be placed as `/etc/logrotate.d/celery`:

```
/var/log/celery/*.log {
        weekly
        missingok
        rotate 12
        compress
        notifempty
}
```

Periodic tasks using Celery beat

Weblate comes with built-in setup for scheduled tasks. You can however define additional tasks in `settings.py`, for example see *Consignas diferidas*.

The tasks are supposed to be executed by Celery beats daemon. In case it is not working properly, it might not be running or its database was corrupted. Check the Celery startup logs in such case to figure out root cause.

Monitoring Celery status

You can find current length of the Celery task queues in the *Interfaz de gestión* or you can use `weblate weblate celery_queues` on the command-line. In case the queue will get too long, you will also get configuration error in the admin interface.

Advertencia: The Celery errors are by default only logged into Celery log and are not visible to user. In case you want to have overview on such failures, it is recommended to configure *Collecting error reports*.

Ver también:

Monitoring Weblate, *¿Cómo puedo comprobar que he configurado mi Weblate adecuadamente?*, Configuration and defaults, Workers Guide, Daemonization, Monitoring and Management Guide, `weblate weblate celery_queues`

2.1.12 Monitoring Weblate

Weblate provides the `/healthz/` URL to be used in simple health checks, for example using Kubernetes. The Docker container has built-in health check using this URL.

For monitoring metrics of Weblate you can use `GET /api/metrics/` API endpoint.

Ver también:

¿Cómo puedo comprobar que he configurado mi Weblate adecuadamente?, *Monitoring Celery status*, Weblate plugin for Munin

2.1.13 Collecting error reports

Weblate, as any other software, can fail. In order to collect useful failure states we recommend to use third party services to collect such information. This is especially useful in case of failing Celery tasks, which would otherwise only report error to the logs and you won't get notified on them. Weblate has support for the following services:

Sentry

Weblate has built-in support for Sentry. To use it, it's enough to set *SENTRY_DSN* in the `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate has built-in support for Rollbar. To use it, it's enough to follow instructions for Rollbar notifier for Python.

In short, you need to adjust `settings.py`:

```
# 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",
}
```

Everything else is integrated automatically, you will now collect both server and client side errors.

Nota: Error logging also includes exceptions that were gracefully handled, but might indicate a problem - such as failed parsing of an uploaded file.

2.1.14 Migrating Weblate to another server

Migrating Weblate to another server should be pretty easy, however it stores data in few locations which you should migrate carefully. The best approach is to stop Weblate for the migration.

Migrating database

Depending on your database backend, you might have several options to migrate the database. The most straightforward approach is to use database native tools, as they are usually the most effective (e.g. **`mysqldump`** or **`pg_dump`**). Alternatively you can use replication in case your database supports it.

Ver también:

Migrating between databases described in *Migrating from other databases to PostgreSQL*.

Migrating VCS repositories

The VCS repositories stored under *DATA_DIR* need to be migrated as well. You can simply copy them or use **rsync** to do the migration more effectively.

Otras notas

No olvide trasladar los otros servicios que Weblate esté utilizando, como Redis, las tareas de Cron o los dorsales de autenticación personalizados.

2.2 Implantaciones de Weblate

Es sencillo instalar Weblate en su nube. Consulte la guía detallada correspondiente a su plataforma:

- *Instalar con Docker*
- *Instalar en OpenShift*
- *Instalación en Kubernetes*

2.2.1 Third-party deployments for Weblate

Nota: Following deployments are not developed or supported by Weblate team. Parts of the setup might vary from what is described in this documentation.

Pila de Weblate para Bitnami

Bitnami provides a Weblate stack for many platforms at <https://bitnami.com/stack/weblate>.

Ver también:

Weblate packaged by Bitnami

Paquete Cloudron de Weblate

Cloudron is a platform for self-hosting web applications. Weblate installed with Cloudron will be automatically kept up-to-date. The package is maintained by the Cloudron team at their Weblate package repo.

Weblate en YunoHost

The self-hosting project YunoHost provides a package for Weblate. Once you have your YunoHost installation, you may install Weblate as any other application. It will provide you with a fully working stack with backup and restoration, but you may still have to edit your settings file for specific usages.

Utilice su interfaz administrativa o pulse en este botón (que le llevará a su servidor):

También se puede utilizar la interfaz de línea de órdenes:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Actualizar Weblate

2.3.1 Docker image upgrades

The official Docker image (see *Instalar con Docker*) has all Weblate upgrade steps integrated. There are typically no manual steps needed besides pulling latest version.

Ver también:

Actualizar el contenedor de Docker

2.3.2 Instrucciones de actualización genéricas

Before upgrading, please check the current *Requisitos de software* 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 *Instrucciones específicas de las versiones* 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.

Nota: 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 *Respaldar y trasladar Weblate*.

---

1. Stop wsgi and Celery processes. The upgrade can perform incompatible changes in the database, so it is always safer to avoid old processes running while upgrading.
2. Upgrade Weblate code.

   For pip installs it can be achieved by:

   ```
   pip install -U "Weblate[all]==version"
   ```

   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 '.[all]'
   # 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 *Dependencias opcionales*, please check if they cover features you want.

4. Upgrade configuration file, refer to `settings_example.py` or *Instrucciones específicas de las versiones* for needed steps.
5. Upgrade database structure:

```
weblate migrate --noinput
```

6. Collect updated static files (see *Running server* and *Serving static files*):

```
weblate collectstatic --noinput --clear
```

7. Compress JavaScript and CSS files (optional, see *Compressing client assets*):

```
weblate compress
```

8. If you are running version from Git, you should also regenerate locale files every time you are upgrading. You can do this by invoking:

```
weblate compilemessages
```

9. Verify that your setup is sane (see also *Puesta en marcha de entorno de producción*):

```
weblate check --deploy
```

10. Reinicie Celery (ver *Tareas en segundo plano con Celery*).

2.3.3 Instrucciones específicas de las versiones

Upgrade from 2.x

If you are upgrading from 2.x release, always first upgrade to 3.0.1 and then continue upgrading in the 3.x series. Upgrades skipping this step are not supported and will break.

Ver también:

Upgrade from 2.20 to 3.0 in Weblate 3.0 documentation

Upgrade from 3.x

If you are upgrading from 3.x release, always first upgrade to 4.0.4 or 4.1.1 and then continue upgrading in the 4.x series. Upgrades skipping this step are not supported and will break.

Ver también:

Upgrade from 3.11 to 4.0 in Weblate 4.0 documentation

Upgrade from 4.0 to 4.1

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- There are several changes in `settings_example.py`, most notable middleware changes, please adjust your settings accordingly.
- There are new file formats, you might want to include them in case you modified the *WEBLATE_FORMATS*.
- There are new quality checks, you might want to include them in case you modified the *CHECK_LIST*.
- There is change in `DEFAULT_THROTTLE_CLASSES` setting to allow reporting of rate limiting in the API.
- There are some new and updated requirements.

- There is a change in `INSTALLED_APPS`.
- 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.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.1 to 4.2

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- Upgrade from 3.x releases is not longer supported, please upgrade to 4.0 or 4.1 first.
- There are some new and updated requirements.
- 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.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.2 to 4.3

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- There are some changes in quality checks, you might want to include them in case you modified the *CHECK_LIST*.
- The source language attribute was moved from project to a component what is exposed in the API. You will need to update *Cliente de Weblate* in case you are using it.
- The database migration to 4.3 might take long depending on number of strings you are translating (expect around one hour of migration time per 100,000 source strings).
- There is a change in `INSTALLED_APPS`.
- There is a new setting *SESSION_COOKIE_AGE_AUTHENTICATED* which complements `SESSION_COOKIE_AGE`.
- In case you were using **`hub`** or **`lab`** to integrate with GitHub or GitLab, you will need to reconfigure this, see *GITHUB_CREDENTIALS* and *GITLAB_CREDENTIALS*.

Distinto en la versión 4.3.1:

- The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

Distinto en la versión 4.3.2:

- The `post_update` method of add-ons now takes extra `skip_push` parameter.

Ver también:

Instrucciones de actualización genéricas

Actualizar de la 4.3 a la 4.4

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- There is a change in `INSTALLED_APPS`, `weblate.configuration` has to be added there.
- Ahora es obligatorio el uso de Django 3.1.
- In case you are using MySQL or MariaDB, the minimal required versions have increased, see *MySQL y MariaDB*.

Distinto en la versión 4.4.1:

- *gettext monolingüe* 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 *`weblate weblate loadpo`*.
- Increased minimal required version of translate-toolkit to address several file format issues.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.4 to 4.5

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- The migration might take considerable time if you had big glossaries.
- Los glosarios ahora se almacenan como componentes normales.
- The glossary API is removed, use regular translation API to access glossaries.
- There is a change in `INSTALLED_APPS` - `weblate.metrics` should be added.

Distinto en la versión 4.5.1:

- There is a new dependency on the *pyahocorasick* module.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.5 a la 4.6

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- There are new file formats, you might want to include them in case you modified the *`WEBLATE_FORMATS`*.
- La aplicación para crear componentes ahora usa automáticamente *URL internos de Weblate*, consulte *`POST /api/projects/(string:project)/components/`*.
- There is a change in dependencies and `PASSWORD_HASHERS` to prefer Argon2 for passwords hashing.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.6 a la 4.7

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- 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.
- Ahora se requiere Django 3.2.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.7 a la 4.8

Please follow *Instrucciones de actualización genéricas* in order to perform update.

There are no additional upgrade steps needed in this release.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.8 a la 4.9

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- There is a change in storing metrics, the upgrade can take long time on larger sites.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.9 a la 4.10

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- There is a change in per-project groups, the upgrade can take long time on sites with thousands of projects.
- Django 4.0 has made some incompatible changes, see Backwards incompatible changes in 4.0. Weblate still supports Django 3.2 for now, in case any of these are problematic. Most notable changes which might affect Weblate:
  - 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.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.10 a la 4.11

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- Weblate ahora requiere Python 3.7 o posterior.
- The implementation of *Gestionar el control de acceso por proyecto* 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`).

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.11 a la 4.12

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- No se requieren pasos especiales.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.12 a la 4.13

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- The *Definiciones de idioma* are now automatically updated on upgrade, use *UPDATE_LANGUAGES* to disable that.
- Handling of context and location has been changed for *Archivos RC de Windows*, *Archivos HTML*, *Formato IDML*, and *Archivos de texto* 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.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.13 a la 4.14

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- 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*.
- **Changed in 4.14.2:** Deprecated insecure configuration of VCS service API keys via _TOKEN/_USERNAME configuration instead of _CREDENTIALS list. In Docker, please add matching _HOST directive. For example see *WEBLATE_GITHUB_HOST* and *GITHUB_CREDENTIALS*.

Ver también:

Instrucciones de actualización genéricas

Actualizar desde la 4.14 a la 4.15

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- Weblate now requires `btree_gin` extension in PostgreSQL. The migration process will install it if it has sufficient privileges. See *Crear una base de datos en PostgreSQL* 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 make 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.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.15 to 4.16

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- Celery beat is now storing the tasks schedule in the database, `CELERY_BEAT_SCHEDULER` and `INSTALLED_APPS` need to be changed for that.
- The deprecated VCS setting for credentials is no longer supported, see *Actualizar desde la 4.13 a la 4.14*.
- Upgrade of *django-crispy-forms* requires changes in `INSTALLED_APPS`.
- Integration of *django-cors-headers* requires changes in `INSTALLED_APPS` and `MIDDLEWARE`.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.16 to 4.17

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- Migration to rewritten metrics storage might take considerable time on larger Weblate instance (expect around 15 minutes per GB of `metrics_metric` table). To reduce downtime, you can copy `weblate/metrics/migrations/*.py` from Weblate 4.17 to 4.16 and start the migration in the background. Once it is completed, perform full upgrade as ususal.
- Docker container now requires PostgreSQL 12 or newer, please see *Actualización del contenedor de PostgreSQL* for upgrade instructions. Weblate itself supports older versions as well, when appropriate Django version is installed.

Advertencia: Migration on MySQL will try to load all metrics into memory due to limitation of the Python database driver. You might need to prune metrics prior to migration if you want to continue using MySQL. Please consider switching to PostgreSQL, see *Migrating from other databases to PostgreSQL*.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.17 to 4.18

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- Dropped support for PostgreSQL 10, 11, MySQL 5.7 and MariaDB 10.2, 10.3.
- Dropped support for Python 3.7.
- The *Fluent format* format changed identification of some strings, you might need to force reloading of the translation files to see the changes.
- There are several changes in `settings_example.py`, most notable is change in `COMPRESS_OFFLINE_CONTEXT`, please adjust your settings accordingly.

Ver también:

Instrucciones de actualización genéricas

2.3.4 Upgrading from Python 2 to Python 3

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 Migrating from other databases to PostgreSQL

If you are running Weblate on other dabatase 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.

Crear una base de datos en PostgreSQL

Suele ser una buena idea ejecutar Weblate en su propia base de datos, en una cuenta de usuario separada:

```
# 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

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Migrating using Django JSON dumps

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 *Migrar a PostgreSQL con 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",
```

(continué en la próxima página)

(proviene de la página anterior)

```
        # 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.

Migrar a PostgreSQL con 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 Migrar de Pootle

As Weblate was originally written as replacement from Pootle, it is supported to migrate user accounts from Pootle. You can dump the users from Pootle and import them using *weblate weblate importusers*.

2.4 Respaldar y trasladar Weblate

2.4.1 Copias de seguridad a nivel del proyecto

Nuevo en la versión 4.14.

Advertencia: Restaurar las copias de seguridad solo es compatible cuando se usa PostgreSQL o MariaDB 10.5+ como base de datos.

El proyecto crea copias de seguridad de todo el contenido de la traducción en Weblate (proyectos, componentes, traducciones, comentarios de las cadenas, sugerencias o comprobaciones). También es posible transferir entre proyectos de Weblate.

Se puede realizar una copia de seguridad del proyecto en *Manage* ↓ *Backups*. La copia de seguridad se puede restaurar al crear un proyecto (véase *Añadir proyectos y componentes de traducción*).

Actualmente las copias de seguridad no incluyen la información de control de acceso ni el historial.

Los comentarios y las sugerencias se respaldan junto con el nombre de usuario de quién los creó. Al importar se le asigna al usuario correspondiente. Si no se encuentra el nombre de usuario, se le asigna a un usuario anónimo.

Las copias de seguridad se guardan en el servidos que se configuró como *PROJECT_BACKUP_KEEP_DAYS* y :setting:`PROJECT_BACKUP_KEEP_COUNT (por defecto se guardan hasta 3 copias de seguridad por 30 días).

2.4.2 Copia de respaldo automatizada utilizando BorgBAckup

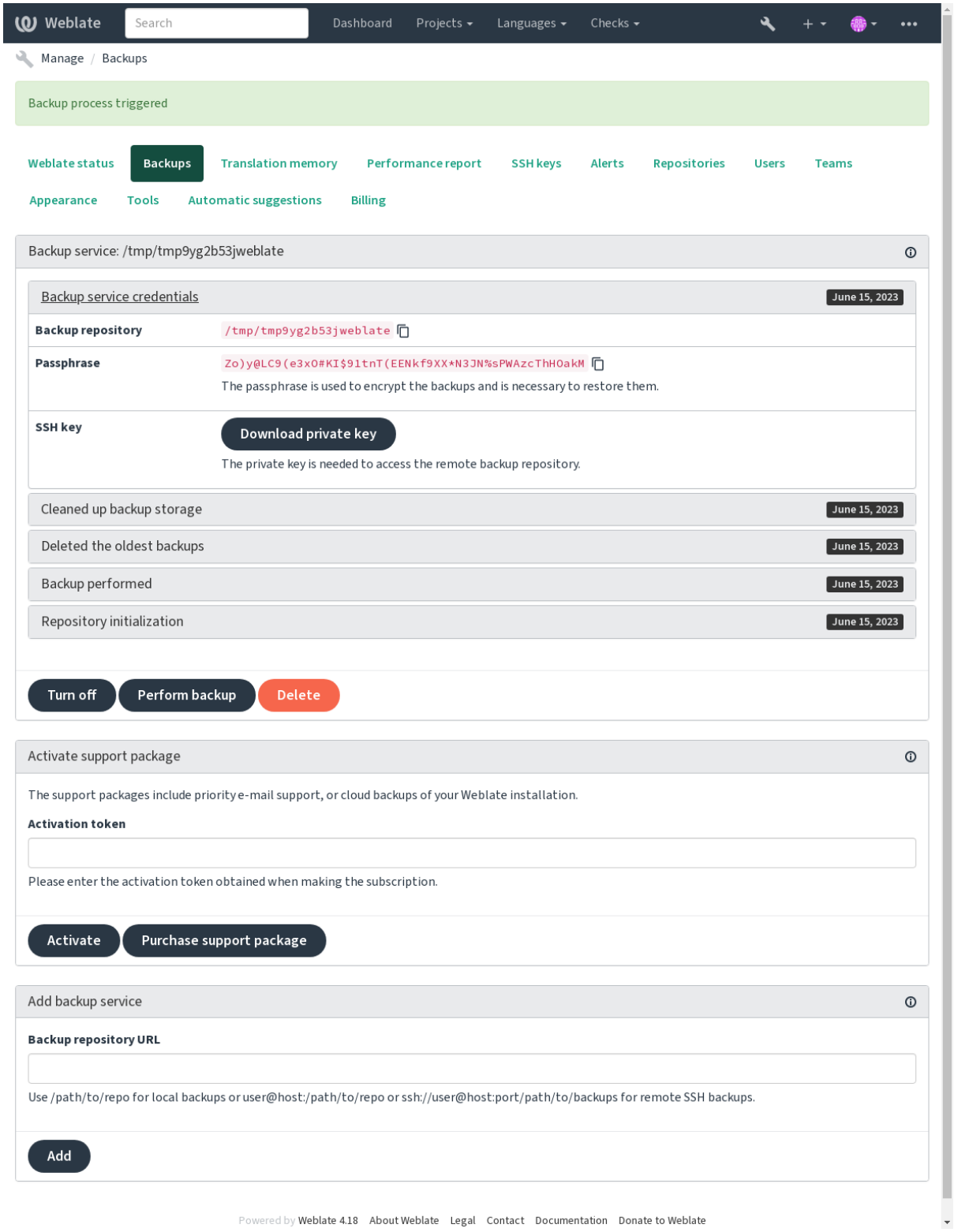
Nuevo en la versión 3.9.

Weblate tiene soporte integrado para crear copias de seguridad de servicios usando BorgBackup. Borg crea copias de seguridad cifradas que ocupan poco espacio y que se pueden almacenar de forma segura en la nube. Las copias de seguridad se pueden controlar en la interfaz de gestión desde la pestaña *Backups*.

Distinto en la versión 4.4.1: Se incluyen las bases de datos de tanto PostgreSQL como MySQL/MariaDB en las copias de respaldo automatizadas.

Las copias de seguridad que utilizan Borg son incrementales y Weblate está configurado para mantener las siguientes copias de seguridad:

- Copias de respaldo diarias para 14 días
- Copias de respaldo semanales para 8 semanas
- Copias de respaldo mensuales para 6 meses


Weblate
Search
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/tmp9yg2b53jweblate
Backup service credentials
June 15, 2023
Backup repository
/tmp/tmp9yg2b53jweblate
Passphrase
Zo)y@LC9(e3xO#KI$91tnT(EENkf9XX*N3JN%sPWAzcThHOakM
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.
Cleaned up backup storage
June 15, 2023
Deleted the oldest backups
June 15, 2023
Backup performed
June 15, 2023
Repository initialization
June 15, 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 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Clave de cifrado de Borg

BorgBackup crea copias de seguridad cifradas y no podrá restaurarlas sin la frase de contraseña. La frase de contraseña se genera al agregar un nuevo servicio de respaldo y debe copiarla y guardarla en un lugar seguro.

Si utilizas *Almacenamiento de copia de seguridad proporcionado por Weblate*, haz una copia de seguridad de tu clave SSH privada también, ya que se utiliza para acceder a tus copias de seguridad.

Ver también:

borg init

Personalización de la copia de seguridad

- La copia de seguridad de la base de datos se puede configurar mediante `DATABASE_BACKUP`.
- La creación de copias de seguridad puede personalizarse mediante `BORG_EXTRA_ARGS`.

2.4.3 Almacenamiento de copia de seguridad proporcionado por Weblate

La forma más sencilla de hacer una copia de seguridad de su instancia de Weblate es adquirir el servicio de copia de seguridad en weblate.org. Así es como se pone en marcha:

1. Compre el *Servicio de copia de seguridad* en https://welate.org/support/#backup.
2. Introduzca la clave obtenida en la interfaz de gestión, consulte *Integrating support*.
3. Weblate se conecta al servicio en la nube y obtiene información de acceso para las copias de seguridad.
4. Active la nueva configuración de las copias de seguridad desde la pestaña *Backups*.
5. Haga una copia de seguridad de sus credenciales de Borg para poder restaurar las copias de seguridad, consulte *Clave de cifrado de Borg*.

Consejo: El paso manual de encender todo está ahí para su seguridad. Sin tu consentimiento no se envía ningún dato al repositorio de copias de seguridad obtenido a través del proceso de registro.

2.4.4 Utilizar un almacenamiento personalizado para los respaldos

También puede utilizar su propio almacenamiento para las copias de seguridad. Se puede utilizar SSH para almacenar las copias de seguridad en el destino remoto, el servidor de destino necesita tener instalado BorgBackup.

Ver también:

General en la documentación de Borg

Sistema de archivos local

Es recomendable especificar una ruta absoluta para la copia de respaldo local, como */ruta/al/respaldo*. El directorio debe ser escribible por la cuenta de usuario que ejecute Weblate (vea *Permisos del sistema de archivos*). Si no existe la ubicación, Weblate intentará crearla, pero necesita permiso para hacerlo.

Consejo: Siempre que se ejecute Weblate en Docker, hay que asegurarse de que la ubicación de las copias de respaldo esté expuesta como volumen desde el contenedor de Weblate. De otro modo, Docker descartará las copias de respaldo al momento de reiniciar el contenedor.

Una opción es colocar las copias de seguridad en un volumen existente, por ejemplo `/app/data/borgbackup`. Este es un volumen existente en el contenedor.

También puede añadir un nuevo contenedor para las copias de seguridad en el archivo Docker Compose, por ejemplo, utilizando `/borgbackup`:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

El propietario del directorio donde se habrán de almacenar las copias de respaldo debe ser el UID 1000, o Weblate no podrá guardar las copias de respaldo allí.

Copias de respaldo remotas

Para crear copias de seguridad remotas, tendrá que instalar BorgBackup en otro servidor que sea accesible para su implementación de Weblate a través de SSH utilizando la clave SSH de Weblate:

1. Prepare un servidor donde se almacenarán sus copias de seguridad.
2. Instala el servidor SSH en él (lo tendrás por defecto con la mayoría de las distribuciones de Linux).
3. Instale BorgBackup en ese servidor; la mayoría de las distribuciones de Linux tienen paquetes disponibles (véase Installation).
4. Elija un usuario existente o cree un nuevo usuario que se utilizará para las copias de seguridad.
5. Añade la clave SSH de Weblate al usuario para que Weblate pueda SSH al servidor sin necesidad de contraseña (ver *Clave SSH de Weblate*).
6. Configure la ubicación de la copia de seguridad en Weblate como `usuario@host:/ruta/a/backups` o `ssh://usuario@host:puerto/ruta/a/backups`.

Consejo: *Almacenamiento de copia de seguridad proporcionado por Weblate* le proporciona copias de seguridad remotas automatizadas sin ningún esfuerzo.

Ver también:

Clave SSH de Weblate, General

2.4.5 Restaurar a partir de BorgBackup

1. Restablece el acceso a tu repositorio de copias de seguridad y prepara tu frase de acceso a las mismas.
2. Listar todas las copias de seguridad en el servidor usando `borg list REPOSITORY`.
3. Restaura la copia de seguridad deseada en el directorio actual utilizando `borg extract REPOSITORY::ARCHIVE`.
4. Restaure la base de datos desde el volcado de SQL colocado en el directorio `backup` en el directorio de datos de Weblate (ver *Datos volcados para las copias de respaldo*).
5. Copie la configuración de Weblate (`backups/settings.py`, consulte *Datos volcados para las copias de respaldo*) en la ubicación correcta, consulte *Adjusting configuration*.

   Cuando se utiliza un contenedor Docker, el archivo de configuración ya está incluido en el contenedor y debe restaurar las variables de entorno originales. El archivo `environment.yml` podría ayudarte con esto (ver *Datos volcados para las copias de respaldo*).
6. Copiar todo el directorio de datos restaurado en la ubicación configurada por *DATA_DIR*.

   Si utiliza un contenedor Docker, coloque los datos en el volumen de datos, véase *Volúmenes de contenedores Docker*.

Por favor, asegúrese de que los archivos tienen la propiedad y los permisos correctos, véase *Permisos del sistema de archivos*.

La sesión de Borg podría verse así:

```
$ 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:
```

Ver también:

borg list, borg extract

2.4.6 Copia de respaldo manual

En función de lo que desee guardar, respalde los tipos de datos que Weblate almacena en cada sitio respectivo.

Consejo: Si estás haciendo las copias de seguridad manuales, puede que quieras silenciar el aviso de Weblate sobre la falta de copias de seguridad añadiendo `weblate.I028` a `SILENCED_SYSTEM_CHECKS` en `settings.py` o *`WEBLATE_SILENCED_SYSTEM_CHECKS`* para Docker.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

Base de datos

La ubicación real del almacenamiento depende de la configuración de su base de datos.

Consejo: El almacenamiento más importante es el de la base de datos. Configure copias de respaldo periódicas de la base de datos. Sin esta, todas las traducciones desaparecerán.

Copia de seguridad de la base de datos nativa

El enfoque recomendado es guardar un volcado de la base de datos utilizando herramientas nativas de la base de datos como **`pg_dump`** o **`mysqldump`**. Suele funcionar mejor que la copia de seguridad de Django, y restaura tablas completas con todos sus datos.

You can restore this backup in a newer Weblate release, it will perform all the necessary migrations when running in *`weblate weblate migrate`*. Please consult *Actualizar Weblate* on more detailed info on how to upgrade between versions.

Copia de seguridad de la base de datos de Django

Alternativamente, puedes hacer una copia de seguridad de tu base de datos utilizando el comando `dumpdata` de Django. De esta manera la copia de seguridad es agnóstica a la base de datos y puede ser utilizada en caso de que quieras cambiar el backend de la base de datos.

Prior to restoring the database you need to be running exactly the same Weblate version the backup was made on. This is necessary as the database structure does change between releases and you would end up corrupting the data in some way. After installing the same version, run all database migrations using *`weblate weblate migrate`*.

Después, algunas entradas ya estarán creadas en la base de datos y las tendrá también en la copia de seguridad de la base de datos. Lo recomendable es eliminar dichas entradas manualmente utilizando el shell de gestión (ver *Invocar órdenes de gestión*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Archivos

Si tiene suficiente espacio para hacer copias de seguridad, simplemente haga una copia de seguridad de todo el *DATA_DIR*. Esto es una apuesta segura incluso si incluye algunos archivos que no quieres. Las siguientes secciones describen en detalle lo que debes respaldar y lo que puedes omitir.

Datos volcados para las copias de respaldo

Distinto en la versión 4.7: El volcado del entorno se añadió como `environment.yml` para ayudar en la restauración en los entornos Docker.

Almacenados en *DATA_DIR* `/backups`.

Weblate vuelca varios datos aquí, y puedes incluir estos archivos para obtener copias de seguridad más completas. Los archivos se actualizan diariamente (se requiere un servidor de beats Celery en funcionamiento, véase *Tareas en segundo plano con Celery*). Actualmente, esto incluye:

- Configuración de Weblate como `settings.py` (también hay una versión ampliada en `settings-expanded.py`).
- Copia de seguridad de la base de datos PostgreSQL como `database.sql`.
- Volcado del entorno como `environment.yml`.

Las copias de seguridad de la base de datos se guardan como texto plano por defecto, pero también pueden comprimirse o saltarse por completo utilizando *DATABASE_BACKUP*.

Para restaurar la copia de seguridad de la base de datos, cárguela utilizando las herramientas de la base de datos, por ejemplo:

```
psql --file=database.sql weblate
```

Repositorios de control de versiones

Almacenado en *DATA_DIR* `/vcs`.

Los repositorios de control de versiones contienen una copia de tus repositorios upstream con los cambios de Weblate. Si tienes *Enviar al consignar* activado para todos tus componentes de traducción, todos los cambios de Weblate se incluyen en el flujo ascendente. No es necesario hacer una copia de seguridad de los repositorios en el lado de Weblate, ya que se pueden clonar de nuevo desde la(s) ubicación(es) de subida sin pérdida de datos.

Claves SSH y GPG

Almacenado en *DATA_DIR* `/ssh` y *DATA_DIR* `/home`.

Si utiliza las claves SSH o GPG que Weblate genera, debe realizar copias de respaldo de esas ubicaciones. De lo contrario, podría perder las claves privadas y habrá de generar nuevas.

Archivos cargados por los usuarios

Almacenado en *DATA_DIR* `/media`.

Debe crear copias de respaldo de todos los archivos que cargan los usuarios (p. ej., *Contexto visual para cadenas*).

Tareas de Celery

La cola de tareas de Celery puede contener alguna información, pero normalmente no es necesaria para una copia de seguridad. A lo sumo se perderán las actualizaciones que aún no han sido procesadas en la memoria de traducción. Se recomienda realizar la actualización del texto completo o del repositorio en el momento de la restauración, por lo que no hay problema en perderlos.

Ver también:

Tareas en segundo plano con Celery

Órdenes de interfaz de texto para efectuar copias de respaldo manualmente

Con la ayuda de una tarea de cron es posible montar una orden de Bash que se ejecute diariamente. Por ejemplo:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz␣
↪backups vcs ssh home media fonts secret
```

La cadena entre comillas después de *XZ_OPT* permite elegir las opciones de xz, por ejemplo la cantidad de memoria utilizada para la compresión; véase https://linux.die.net/man/1/xz

Puede ajustar la lista de carpetas y de archivos para adecuarla a sus necesidades. Para evitar guardar la memoria de traducción (en la carpeta de las copias de respaldo), puede utilizar:

```
$ 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 Restaurar una copia de respaldo manual

1. Restaure todos los datos de los que ha hecho copia de respaldo.
2. Update all repositories using *weblate weblate updategit*.

   ```
   weblate updategit --all
   ```

2.4.8 Trasladar una instalación de Weblate

Para mudar su instalación a un sistema diferente, siga las instrucciones de respaldo y restauración anteriores.

Ver también:

Upgrading from Python 2 to Python 3, *Migrating from other databases to PostgreSQL*

2.5 Autenticación

2.5.1 Registro de usuarios

La configuración por defecto de Weblate es utilizar python-social-auth, un formulario en el sitio web para gestionar el registro de nuevos usuarios. Después de confirmar su correo electrónico, un nuevo usuario puede contribuir o autenticarse utilizando uno de los servicios de terceros.

También puedes desactivar el registro de nuevos usuarios mediante *REGISTRATION_OPEN*.

Los intentos de autenticación están sujetos a *Rate limiting*.

2.5.2 Dorsales de autenticación

La solución incorporada de Django se utiliza para la autenticación, incluyendo varias opciones sociales para hacerlo. Su uso permite importar la base de datos de usuarios de otros proyectos basados en Django (ver *Migrar de Pootle*).

Django también se puede configurar para configurar otros medios.

Ver también:

Configuración de autenticación describe cómo configurar la autenticación en la imagen oficial para Docker.

2.5.3 Autenticación social

Gracias a Welcome to Python Social Auth's documentation!, Weblate admite la autenticación a través de numerosos servicios de terceros, tales como GitLab, Ubuntu y Fedora, entre otros.

Consulta su documentación para obtener instrucciones de configuración genéricas en Django Framework.

Nota: Por defecto, Weblate confía en los servicios de autenticación de terceros para proporcionar una dirección de correo electrónico validada. Si algunos de los servicios que deseas utilizar no lo soportan, por favor, ejecuta la validación del correo electrónico en el lado de Weblate configurando FORCE_EMAIL_VALIDATION para ellos. Por ejemplo:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Ver también:

Pipeline

Habilitar backends individuales es bastante fácil, sólo es cuestión de añadir una entrada al ajuste AUTHENTICATION_BACKENDS y posiblemente añadir las claves necesarias para un método de autenticación determinado. Ten en cuenta que algunos backends no proporcionan el correo electrónico del usuario por defecto, tienes que solicitarlo explícitamente, de lo contrario Weblate no será capaz de acreditar adecuadamente las contribuciones de los usuarios.

Consejo: La mayoría de los backends de autenticación requieren HTTPS. Una vez que el HTTPS esté habilitado en tu servidor web, por favor configura Weblate para que lo informe correctamente usando *ENABLE_HTTPS*, o mediante *WEBLATE_ENABLE_HTTPS* en el contenedor Docker.

Ver también:

Dorsal de Python Social Auth

Autenticación por OpenID

Para servicios basados en OpenID basta con activarlos. En esta sección se describe cómo activar la autenticación por OpenID de OpenSUSE, Fedora y Ubuntu:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.suse.OpenSUSEOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Ver también:

OpenID

Autenticación por GitHub

Tienes que registrar una aplicación OAuth en GitHub y luego decirle a Weblate todos sus secretos:

```
# 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"]
```

The GitHub should be configured to have callback URL as `https://WEBLATE SERVER/accounts/complete/github/`.

There are similar authentication backends for GitHub for Organizations and GitHub for Teams. Their settings are named `SOCIAL_AUTH_GITHUB_ORG_*` and `SOCIAL_AUTH_GITHUB_TEAM_*`, and they require additional setting of the scope - `SOCIAL_AUTH_GITHUB_ORG_NAME` or `SOCIAL_AUTH_GITHUB_TEAM_ID`. Their callback URLs are `https://WEBLATE SERVER/accounts/complete/github-org/` and `https://WEBLATE SERVER/accounts/complete/github-teams/`.

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

GitHub

GitHub EE authentication

You need to register an OAuth App on GitHub EE and then tell Weblate all its secrets:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github_enterprise.GithubEnterpriseOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_ENTERPRISE_KEY = "GitHub OAuth App Client ID"
SOCIAL_AUTH_GITHUB_ENTERPRISE_SECRET = "GitHub OAuth App Client Secret"
SOCIAL_AUTH_GITHUB_ENTERPRISE_URL = "https://git.example.com/"
SOCIAL_AUTH_GITHUB_ENTERPRISE_API_URL = "https://git.example.com/api/v3/"
SOCIAL_AUTH_GITHUB_ENTERPRISE_SCOPE = ["user:email"]
```

The GitHub OAuth App should be configured to have callback URL as `https://WEBLATE_SERVER/accounts/complete/github-enterprise/`.

Instead GitHub OAuth App, GitHub App can also be used. With GitHub App permissions can be granted on repositories, organisation and/or user level. If you decide to use GitHub App, you need to enable *Access: Read-only* permission for Users - <Email addresses> and Organisation - <Members>.

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

GitHub Enterprise

Autenticación por Bitbucket

Tienes que registrar una aplicación en Bitbucket y luego decirle a Weblate todos sus secretos:

```
# 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
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

Bitbucket

Google OAuth 2

Para utilizar OAuth 2 de Google, es necesario registrar una aplicación en <https://console.developers.google.com/> y activar la API de Google+.

La URL de redirección es `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"
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

Google

OAuth 2 de Facebook

Como es habitual con los servicios OAuth 2, tienes que registrar tu aplicación en Facebook. Una vez hecho esto, puedes configurar Weblate para utilizarla:

La URL de redirección es `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"]
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

Facebook

OAuth 2 de GitLab

Para utilizar GitLab OAuth 2, es necesario registrar una aplicación en <https://gitlab.com/profile/applications>.

La URL de redirección es `https://WEBLATE SERVER/accounts/complete/gitlab/` y asegúrate de marcar el ámbito *read_user*.

```
# 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/'
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

GitLab

Active Directory de Microsoft Azure

Puede configurarse Weblate para utilizar inquilinos comunes o específicos para la autenticación.

La URL redireccionada es `https://WEBLATE SERVER/accounts/complete/azuread-oauth2/` para las actividades comunes y``https://WEBLATE SERVER/accounts/complete/azuread-tenant-oauth2/`` para la autenticación específica del usuario.

You will need following:

- *Application (client) ID* can be obtained from application page. *Object ID* is not used in Weblate.
- *Directory (tenant) ID* is needed for tenant scoped authentication, what is usually desired.
- *Secret value* is displayed once you generate a secret for an application. *Secret ID* is not used in Weblate.

```
# 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",
)

# Application (client) ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
# Secret value
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Directory (tenant) ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

Microsoft Azure Active Directory

Slack

Para utilizar Slack OAuth 2, necesitas registrar una aplicación en <https://api.slack.com/apps>.

La URL redireccionada es `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 = ""
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver *Set correct site domain*.

Ver también:

Slack

Anulación de los nombres e iconos de los métodos de autenticación

Puedes anular el nombre y el icono del método de autenticación utilizando la configuración como `SOCIAL_AUTH_<NAME>_IMAGE` y `SOCIAL_AUTH_<NAME>_TITLE`. Por ejemplo, el nombre de anulación para Auth0 se vería como:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Desactivar la autenticación por contraseña

La autenticación por correo electrónico y contraseña puede desactivarse eliminando `social_core.backends.email.EmailAuth` desde `AUTHENTICATION_BACKENDS`. Mantén siempre `weblate.accounts.auth.WeblateUserBackend`, es necesaria para la funcionalidad principal de Weblate.

Deshabilitar la autenticación por correo electrónico deshabilitará todas las funciones relacionadas con este: invitación de usuario o la función para restablecer la contraseña.

Truco: Todavía puede usar la autenticación de contraseña para la interfaz de administración, para los usuarios que crea allí manualmente. Simplemente navegue a `/admin/login/`.

Por ejemplo, la autenticación utilizando sólo el proveedor Open ID de openSUSE se puede lograr con lo siguiente:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Autenticación por contraseña

Por defecto `settings.py` viene con un set razonable de `AUTH_PASSWORD_VALIDATORS`:

- Las contraseñas no deben asemejarse demasiado a otros datos personales suyos.
- Las contraseñas deben tener por lo menos 10 caracteres de longitud.
- Las contraseñas no pueden figurar entre las más comúnmente utilizadas.
- Las contraseñas no pueden consistir enteramente de números.
- Las contraseñas no pueden consistir de un único carácter o de solo espacios.
- Las contraseñas no pueden coincidir con alguna que haya utilizado en el pasado.

Puede personalizar esta configuración para que se ajuste a su normativa de contraseñas.

Además, puedes instalar django-zxcvbn-password que ofrece estimaciones bastante realistas de la dificultad de las contraseñas y permite rechazar las contraseñas por debajo de un determinado umbral.

2.5.5 Autenticación por SAML

Nuevo en la versión 4.1.1.

Por favor, sigue las instrucciones de Python Social Auth para la configuración. Diferencias notables:

- Weblate soporta un único IDP que se debe llamar `weblate` en `SOCIAL_AUTH_SAML_ENABLED_IDPS`.
- La URL de los metadatos XML de SAML es `/accounts/metadata/saml/`.
- Los siguientes ajustes se rellenan automáticamente: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Ejemplo de configuración:

```
# 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 ... 8Bbnl+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"
}
```

La configuración por defecto extrae los detalles del usuario de los siguientes atributos, configura tu IDP para proporcionarlos:

Atributo	Referencia de URI de SAML
Nombre completo	`urna:oid:2.5.4.3`
Nombre	`urna:oid:2.5.4.42`
Apellidos	`urna:oid:2.5.4.4`
Correo electrónico	`urna:oid:0.9.2342.19200300.100.1.3`
Nombre de usuario	`urna:oid:0.9.2342.19200300.100.1.1`

Consejo: El ejemplo anterior y la imagen de Docker definen un IDP llamado `weblate`. Es posible que deba configurar esta cadena como *Relay* en su IDP.

Ver también:

Configuración de SAML en el Docker, SAML

2.5.6 Autenticación LDAP

La autenticación LDAP puede lograrse mejor utilizando el paquete *django-auth-ldap*. Puedes instalarlo por los medios habituales:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

Consejo: Este paquete está incluido en el contenedor Docker, véase *Instalar con Docker*.

Nota: Hay algunas incompatibilidades en el módulo LDAP 3.1.0 de Python, que podrían impedirte utilizar esa versión. Si se produce el error AttributeError: "module" object has no attribute "_trace_level", podría ayudar la actualización de python-ldap a la versión 3.0.0.

Una vez que tengas el paquete instalado, lo puedes enlazar a la autenticación de Django:

```
# 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
```

Nota: You should remove `'social_core.backends.email.EmailAuth'` from the AUTHENTICATION_BACKENDS setting, otherwise users will be able to set their password in Weblate, and authenticate using that. Keeping `'weblate.accounts.auth.WeblateUserBackend'` is still needed in order to make permissions and facilitate anonymous users. It will also allow you to sign in using a local admin account, if you have created it (e.g. by using *weblate weblate createadmin*).

Uso de la contraseña de enlace

Si no puedes usar el enlace directo para la autenticación, tendrás que utilizar la búsqueda y proporcionar un usuario para el enlace de la búsqueda. Por ejemplo:

```
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)"
)
```

Integración con Active Directory

```
import ldap
from django_auth_ldap.config import LDAPSearch, NestedActiveDirectoryGroupType

AUTH_LDAP_BIND_DN = "CN=ldap,CN=Users,DC=example,DC=com"
AUTH_LDAP_BIND_PASSWORD = "password"

# User and group search objects and types
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "CN=Users,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s)"
)

# Make selected group a superuser in Weblate
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "OU=Groups,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True
```

Ver también:

Django Authentication Using LDAP, Authentication

2.5.7 Autenticación CAS

Se puede implantar una autenticación CAS al utilizar un paquete como *django-cas-ng*.

El primer paso consiste en revelar el campo Correo electrónico del usuario mediante CAS. Esto debe configurarse en el propio servidor CAS, y necesitará ejecutar al menos la versión 2 de CAS, ya que CAS v1 no admite atributos.

El segundo paso será actualizar Weblate para que utilice el servidor y los atributos de CAS.

Para instalar *django-cas-ng*:

```
pip install django-cas-ng
```

Una vez que haya instalado el paquete, puede conectarlo con el sistema de autenticación de Django; para ello, modifique el archivo `settings.py`:

```
# 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")
```

Finalmente, se puede utilizar una señal para vincular el campo Correo electrónico y el objeto de usuario. Para que esto funcione, debe importar la señal del paquete *django-cas-ng* y conectar su código con esta señal. Realizar esto en el archivo de configuración puede causar problemas, por lo cual se recomienda ponerlo:

- En el método `django.apps.AppConfig.ready()` de la configuración de su aplicación
- En el archivo `urls.py` del proyecto (cuando no existan modelos)

```
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()
```

Ver también:

` Biblioteca de cliente Django CAS <https://github.com/django-cas-ng/django-cas-ng>`_

2.5.8 Configurar la autenticación de Django de terceros

Generalmente cualquier programa adicional Django funciona con Weblate. Solo hay que seguir las instrucciones del programa adicional. Se recomienda tener el soporte de usuario Weblate instalado.

Ver también:

Autenticación LDAP, *Autenticación CAS*

Normalmente la instalación consiste en agregar un soporte de autenticación en `AUTHENTICATION_BACKENDS` e instalar una aplicación de autenticación (si la hay) en `INSTALLED_APPS`:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Control de acceso

Weblate incluye un sistema de privilegios detallado que permite asignar permisos de usuario en toda la instalación o en un ámbito limitado.

Distinto en la versión 3.0: Hasta Weblate 3.0, el sistema de privilegios se basaba únicamente en el de Django, pero ahora se ha construido específicamente para Weblate. Si utiliza una versión antigua, consulte la documentación que corresponda a esa versión.

2.6.1 Control de acceso simple

Si no está administrando toda la instalación de Weblate y solo tiene acceso para administrar ciertos proyectos (como en *Hosted Weblate <https://hosted.weblate.org/> _*), sus opciones de administración de control de acceso se limitan a las siguientes configuraciones. Si no necesita ninguna configuración compleja, será suficiente para ti.

Control de acceso al proyecto

Nota: Los proyectos con el plan gratuito Libre en Hosted Weblate son siempre *Public*. Puede cambiar al plan de pago si desea restringir el acceso a tu proyecto.

Puede limitar el acceso del usuario a proyectos individuales seleccionando una configuración diferente: guilabel: *Control de acceso*. Las opciones disponibles son:

Público
: Visible para todo el mundo.

  Cualquier usuario autenticado puede contribuir.

  El repositorio VCS podría quedar expuesto a cualquiera.

  Elige esta opción para proyectos de código abierto o cuando tu instancia de Weblate sea privada o esté bloqueada

Protegido
: Visible para todo el mundo.

  Solo determinados usuarios pueden contribuir.

  Solo determinados usuarios pueden acceder al repositorio VCS.

  Elige esta opción para ganar visibilidad, pero manteniendo el control sobre quién puede contribuir.

Privado
: Visible solo por usuarios seleccionados.

  Solo determinados usuarios pueden contribuir.

  Solo determinados usuarios pueden acceder al repositorio VCS.

  Choose this for projects that should not be exposed publicly at all.

Personalizado

Visible solo por usuarios seleccionados.

Solo determinados usuarios pueden contribuir.

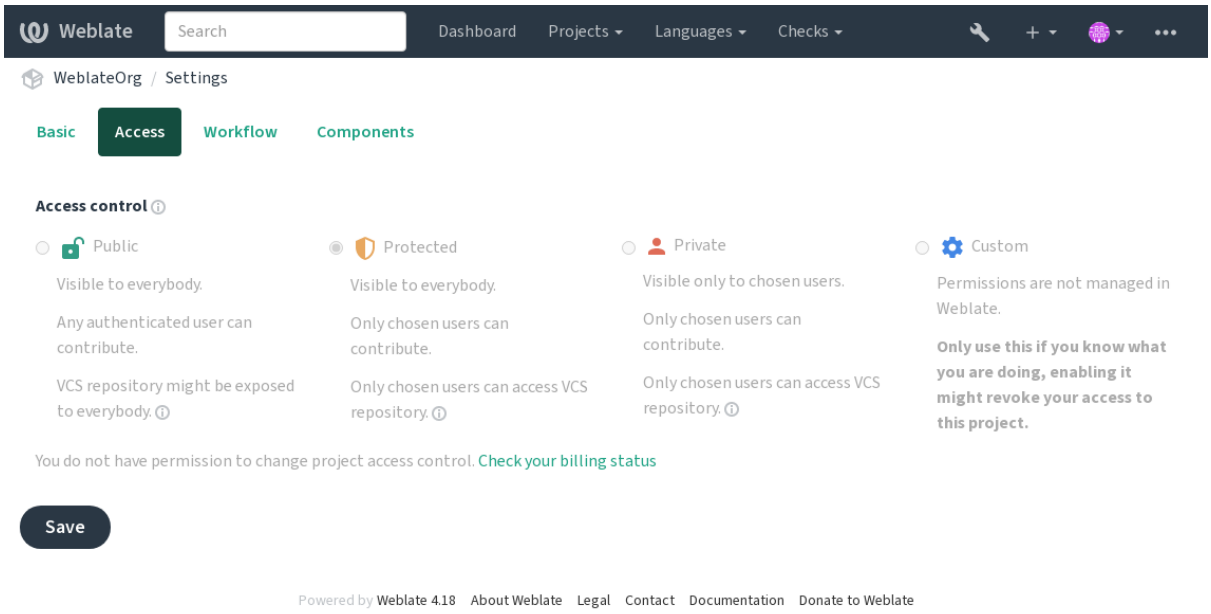
Solo determinados usuarios pueden acceder al repositorio VCS.

No está disponible en Hosted Weblate.

Tendrás que configurar todos los permisos utilizando *Control de acceso personalizado*.

Elije esta opción en tu propia instancia de Weblate si deseas definir el acceso de una forma específica y finamente personalizable.

Access control can be changed in the *Access* tab of the configuration (*Manage* ↓ *Settings*) of each respective project.



El valor predeterminado se puede cambiar mediante :setting: *DEFAULT_ACCESS_CONTROL*.

Nota: Incluso para los proyectos *Privados*, alguna información sobre su proyecto será expuesta: las estadísticas y el resumen de idiomas para toda la instancia incluirán los recuentos de todos los proyectos a pesar de la configuración del control de acceso. El nombre de su proyecto y otra información no pueden ser revelados a través de esto.

Nota: El administrador de la instancia de Weblate puede redefinir el conjunto de permisos disponibles por defecto para los usuarios en los proyectos *Public*, *Protected*, y *Private* utilizando *custom settings*.

Ver también:

Control de acceso

Gestionar el control de acceso por proyecto

Los usuarios con el privilegio *Manage project access* (ver *Lista de privilegios y roles incorporados*) pueden gestionar los usuarios en los proyectos mediante la adición de ellos a los equipos. La colección inicial de equipos es proporcionada por Weblate, pero se pueden definir otros adicionales que proporcionan idiomas y asignarles roles de acceso designados (ver privilegios).

Los siguientes equipos se crean automáticamente para cada proyecto:

Para proyectos *Públicos*, *Protegidos* y *Privados*:

Administración
: Cuenta con todos los permisos disponibles en el proyecto.

Revisión (sólo si *review workflow* está activado)
: Puede aprobar traducciones durante la revisión.

Para proyectos *protegidos* y *privados* únicamente:

Traducir
: Puede traducir el proyecto y cargar traducciones realizadas sin conexión.

Orígenes
: Puede editar las cadenas de origen (si se permite en la *configuración del proyecto*) y la información de la cadena de origen.

Idiomas
: Puede gestionar los idiomas traducibles (añadir o quitar traducciones).

Glosario
: Puede gestionar el glosario (añadir o quitar entradas, o bien cargarlas).

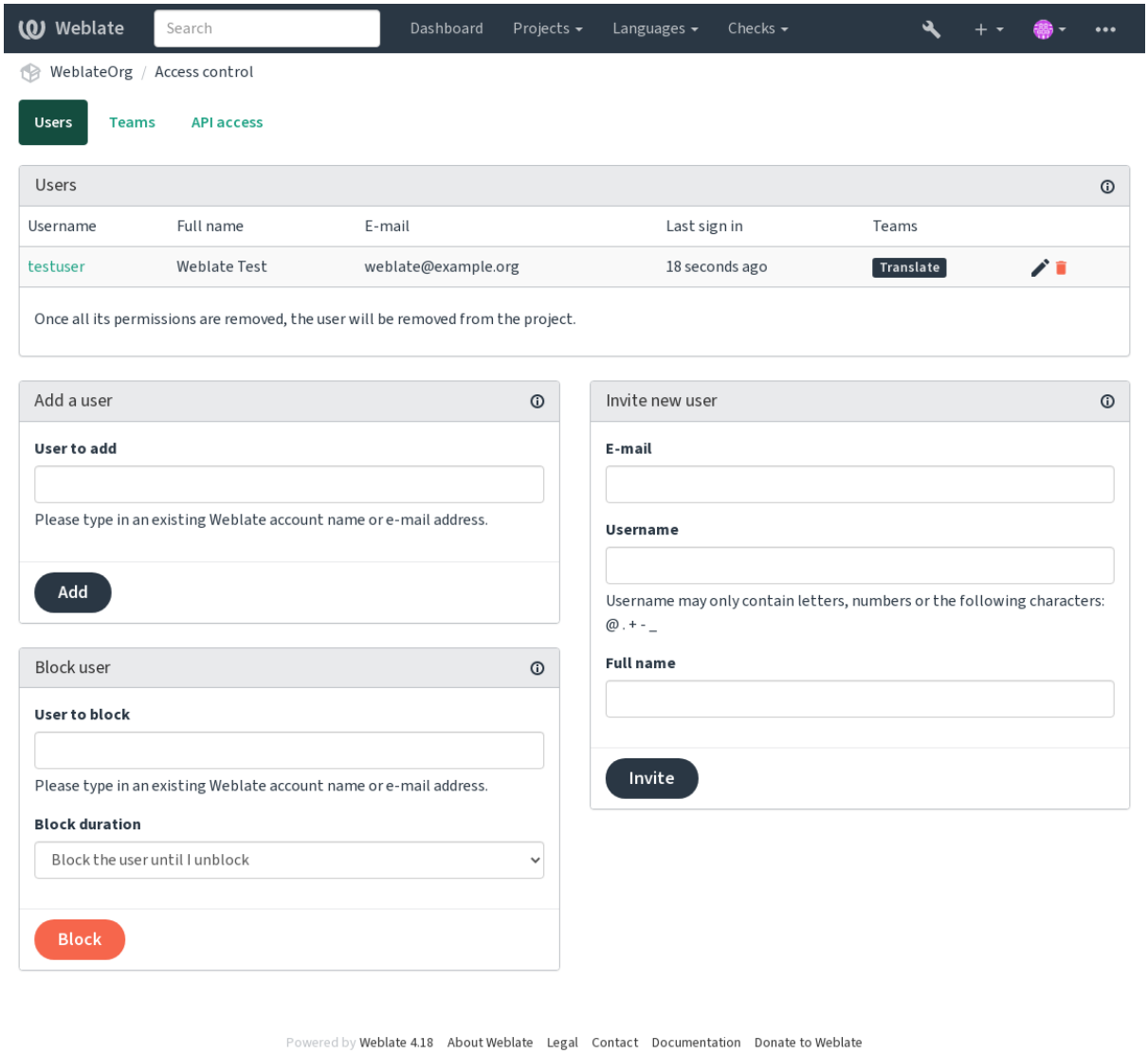
Memoria
: Puede gestionar la memoria de traducción.

Capturas de pantalla
: Puede gestionar las capturas de pantalla (añadirlas o eliminarlas, así como asociarlas a cadenas de origen).

Traducción automática
: Puede usar traducción automática.

Sistema de control de versiones
: Puede gestionar el sistema de control de versiones y acceder al repositorio exportado.

Facturación
: Puede acceder a la información y la configuración de facturación (vea *Facturación*).



Estas funciones están disponibles en la página *Control de acceso*, a la que se puede acceder desde el menú del proyecto *Gestión ↓ Usuarios*.

Administradores del equipo

Nuevo en la versión 4.15.

Cada grupo puede tener un administrador del equipo, el cual puede añadir y quitar usuarios. Esto es útil en caso de querer hacer equipos autogobernados.

Invitación a usuarios nuevos

Además de añadir un usuario existente al proyecto, también es posible invitar a otros nuevos. Cualquier nuevo usuario se creará inmediatamente, pero la cuenta permanecerá inactiva hasta que se registre con un enlace en la invitación enviada por correo electrónico. No es necesario tener ningún privilegio en el sitio para hacerlo, bastaría con un permiso de gestión de acceso en el ámbito del proyecto (por ejemplo, una pertenencia al equipo *Administration*).

Consejo: Si el usuario invitado pierde la validación de la invitación, pueden establecer su contraseña utilizando la dirección de correo electrónico invitado en el formulario de restablecimiento de contraseña, ya que la cuenta ya está creada.

Nuevo en la versión 3.11: Es posible reenviar el correo electrónico para las invitaciones de los usuarios (invalidando cualquier invitación enviada anteriormente).

El mismo tipo de invitaciones están disponibles en todo el sitio desde la *management interface* en la pestaña :*Users*.

Bloquear usuarios

Nuevo en la versión 4.7.

En caso de que algunos usuarios se comporten mal en tu proyecto, tienes la opción de bloquearlos para que no contribuyan. El usuario bloqueado podrá seguir viendo el proyecto si tiene permisos para ello, pero no podrá contribuir.

Gestión de permisos por proyecto

Puede definir sus proyectos como *Protegido* o *Privado* y *gestionar los usuarios* por proyecto en la interfaz de usuario de Weblate.

Por defecto, esto evita que Weblate conceda el acceso proporcionado por *Users* y *Viewers* *default teams* debido a la propia configuración de estos equipos. Esto no impide que se concedan permisos a esos proyectos en todo el sitio modificando los grupos por defecto, creando uno nuevo o creando configuraciones personalizadas adicionales para componentes individuales como se describe en *Control de acceso personalizado* más abajo.

Uno de los principales beneficios de administrar permisos a través de la interfaz de usuario de Weblate es que puedes delegarlo a otros usuarios sin darles el privilegio de superusuario. Para ello, agréguelos al equipo de *Administración* del proyecto.

2.6.2 Control de acceso personalizado

Nota: Esta función no está disponible en Hosted Weblate.

El sistema de permisos está basado en equipos y roles, donde los roles definen un conjunto de permisos, y los equipos los vinculan a los usuarios y a las traducciones, véase *Usuarios, funciones, equipos y permisos* para más detalles.

Las características más potentes del sistema de control de acceso de Weblate por ahora sólo están disponibles a través de la interfaz de administración de *Django admin interface*. Puedes utilizarlo para gestionar los permisos de cualquier proyecto. No tienes que cambiarlo necesariamente a *Custom access control* para usarlo. Sin embargo, debes tener privilegios de superusuario para poder utilizarlo.

Si no estás interesado en los detalles de la implementación, y sólo quieres crear una configuración muy simple basada en los valores predeterminados, o no tienes un acceso a todo el sitio de la instalación de Weblate (como en Hosted Weblate), por favor visita la sección *Control de acceso simple*.

Configuraciones comunes

Esta sección contiene un resumen de algunas configuraciones comunes que pueden interesarte.

Gestión de permisos para todo el sitio

Para gestionar los permisos de toda una instancia a la vez, añade usuarios a los equipos apropiados *default teams*:

- *Users* (esto se hace por defecto mediante la *automatic team assignment*).
- *Reviewers* (si estás utilizando *review workflow* con revisores dedicados).
- *Managers* (si quieres delegar la mayor parte de las operaciones de gestión a otra persona).

Deberías mantener todos los proyectos configurados como *Public* (ver *Control de acceso al proyecto*), de lo contrario los permisos de todo el sitio proporcionados por la pertenencia a los equipos *Users* y *Reviewers* no tendrán ningún efecto.

También puedes conceder algunos permisos adicionales de tu elección a los equipos por defecto. Por ejemplo, puedes querer dar un permiso para gestionar las capturas de pantalla a todos los *Users*.

También puedes definir nuevos equipos personalizados. Si quieres seguir gestionando tus permisos en todo el sitio para estos equipos, elige un valor apropiado para *Project selection* (e.g. *All projects* o *All public projects*).

Permisos personalizados para idiomas, componentes o proyectos

Puedes crear sus propios grupos dedicados para gestionar los permisos de distintos objetos como idiomas, componentes y proyectos. Aunque estos grupos sólo pueden conceder privilegios adicionales, no puede revocar ningún permiso concedido por los grupos a nivel de sitio o proyecto añadiendo otro grupo personalizado.

Ejemplo:

Si quieres (por la razón que sea) permitir la traducción a un idioma específico (digamos *checo*) sólo a un conjunto cerrado de traductores fiables mientras mantienes las traducciones a otros idiomas públicos, tendrás que hacerlo:

1. Elimine el permiso para traducir *checo* de todos los usuarios. En la configuración por defecto esto se puede hacer alterando el grupo *Usuarios* default group.

Tabla 1: Grupo *Usuarios*

Selección de idioma	*Como se ha definido*
Idiomas	Todos menos *checo*

2. Añadir un grupo exclusivo para los traductores de *checo*.

Tabla 2: Grupo *Traductores al checo*

Roles	*Usuario avanzado*
Selección del proyecto	*Todos los proyectos públicos*
Selección de idioma	*Como se ha definido*
Idiomas	*Checo*

3. Añada a este equipo los usuarios a los que desea otorgar permisos.

Como puedes ver, la gestión de permisos de esta manera es poderosa, pero puede ser un trabajo bastante tedioso. No se puede delegar a otro usuario, a no ser que se concedan permisos de superusuario.

Usuarios, funciones, equipos y permisos

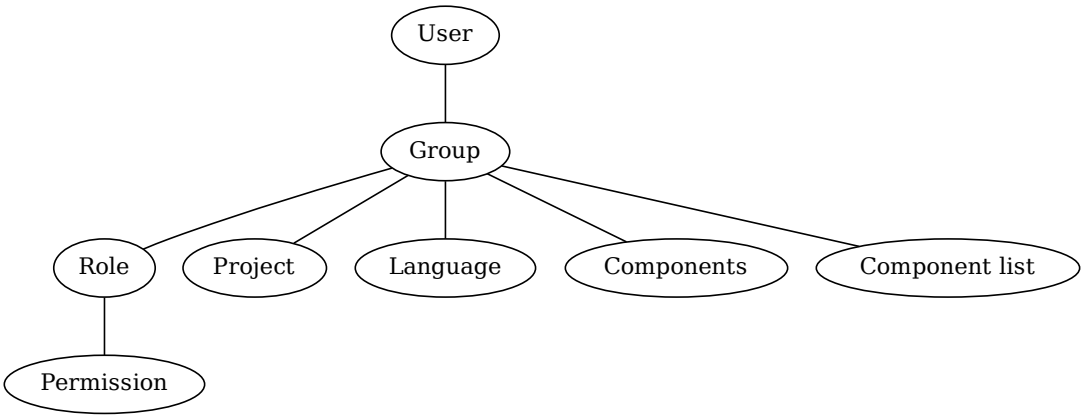
Los modelos de autenticación consisten de varios objetos:

Permiso
Permiso individual definido por Weblate. Los permisos no se pueden asignar a los usuarios. Esto sólo puede hacerse mediante la asignación de roles.

Rol
Un rol define un conjunto de permisos. Esto permite reutilizar este conjunto en varios lugares, facilitando la administración.

Usuario
Un usuario puede ser miembro de varios equipos.

Grupo
El grupo conecta roles, usuarios y objetos de autenticación (proyectos, idiomas y listas de componentes).



Nota: Un grupo puede no tener roles asignados, en ese caso se asume el acceso a navegar por el proyecto por parte de cualquiera (ver debajo).

Acceso a navegar por un proyecto

Un usuario tiene que ser miembro de un grupo vinculado al proyecto, o a cualquier componente dentro de ese proyecto. Tener la membresía es suficiente, no se necesitan permisos específicos para navegar por el proyecto (esto se utiliza en el grupo por defecto *Observadores*, ver default-groups).

Acceso para navegar a un componente

Un usuario puede acceder a los componentes sin restricciones una vez que pueda acceder al proyecto de los componentes (y tendrá todos los permisos que el usuario tenga concedidos para el proyecto). Con *Acceso restringido* activado, el acceso al componente requiere permisos explícitos para el componente (o una lista de componentes en la que esté el componente).

Ámbito de los equipos

El alcance del permiso asignado por los roles en los equipos se aplica mediante las siguientes reglas:

- Si el equipo especifica cualquier *Component list*, todos los permisos otorgados a los miembros de ese equipo se conceden para todos los componentes de las listas de componentes adjuntas al equipo, y se concede un acceso sin permisos adicionales para todos los proyectos en los que estén estos componentes. *Components* y *Projects* se ignoran.
- Si el equpo especifica cualquier *Components*, todos los permisos otorgados a los miembros de ese equipo se conceden para todos los componentes adscritos al equipo, y se concede un acceso sin permisos adicionales para todos los proyectos en los que están estos componentes. *Projects* se ignora.
- De lo contrario, si el equipo especifica cualquier *Proyects*, ya sea enumerándolos directamente o teniendo *Projects selection* fijado en un valor como *All public projects*, todos esos permisos se aplican a todos los proyectos, lo que efectivamente otorga los mismos permisos para acceder a todos los proyectos *unrestricted components*.
- Las restricciones impuestas por la *Languages* de un equipo se aplican por separado, cuando se verifica si un usuario tiene acceso a realizar determinadas acciones. En concreto, se aplica sólo a las acciones directamente relacionadas con el proceso de traducción en sí, como revisar, guardar traducciones, añadir sugerencias, etc.

Consejo: Para automatizar la inclusión de todos los idiomas o proyectos, usa *Language selection* o *Project selection*.

Ejemplo:

Digamos que hay un proyecto `foo` con los componentes: `foo/bar` yd `foo/baz` y el equipo siguiente:

Tabla 3: Grupo *Administradores-Revisores Españoles*

Roles	*Revisar cadenas*, *Gestionar repositorio*
Componentes	foobar
Idiomas	*Español*

Los miembros de ese equipo tendrán los siguientes permisos (asumiendo la configuración de las funciones por defecto):

- Acceso general (navegación) a todo el proyecto `foo` incluyendo los dos componentes que lo componen: `foo/bar` y `foo/baz`.
- Revisar las cadenas en la traducción al español de `foo/bar` (no en otra parte).
- Gestionar VCS para todo el repositorio `foo/bar`, por ejemplo, commit los cambios pendientes realizados por los traductores para todos los idiomas.

Asignación automática de los equipos

En la parte inferior de la página de edición *Group* en la interfaz de administración *Django*, puede especificar *Automatic team assignments*, que es una lista de expresiones regulares utilizadas para asignar automáticamente los usuarios recién creados a un equipo en función de sus direcciones de correo electrónico. Esta asignación sólo se realiza al crear la cuenta.

El caso de uso más común para esta función es asignar todos los usuarios nuevos a un equipo predeterminado. Para ello, probablemente querrá mantener el valor por defecto (`^.*$`) en el campo de expresión regular. Otro caso de uso para esta opción podría ser dar algunos privilegios adicionales a los empleados de su empresa por defecto. Suponiendo que todos ellos utilicen direcciones de correo electrónico corporativas en su dominio, esto se puede conseguir con una expresión como `^.*@miempresa.com`.

Nota: La asignación automática de los equipos a *Usuarios* y *Visores* se recrea siempre al actualizar de una versión de Weblate a otra. Si desea desactivarla, establezca la expresión regular en `^$` (que no coincidirá con nada).

Nota: Por el momento, no hay forma de añadir usuarios existentes a un equipo a través de la interfaz de usuario. Para ello, puede recurrir a la *REST API*.

Equipos y funciones predeterminadas

Tras la instalación, se crea un conjunto de equipos por defecto (véase *Lista de los equipos*).

Estas funciones y equipos se crean en el momento de la instalación. Las funciones incorporadas se mantienen siempre actualizados por la migración de la base de datos al actualizar. En realidad no puedes cambiarlos, por favor define una nueva función si quieres definir tu propio conjunto de permisos.

Lista de privilegios y roles incorporados

Alcance	Permiso	Roles
Facturación (vea *Facturación*)	Ver datos de facturación	*Administración*, *Facturación*
Cambios	Descargar cambios	*Administración*
Comentarios	Enviar comentario	*Administración*, *Editar fuente*, *Usuario ava*
	Eliminar comentario	*Administración*
	Resolver el comentario	*Administración*, *Revisar cadenas*
Componente	Editar configuración de los componentes	*Administración*
	Bloquear componente, impidiendo traducciones	*Administración*
Glosario	Añadir palabra al glosario	*Administración*, *Gestionar glosario*, *Usuari*
	Editar palabra del glosario	*Administración*, *Gestionar glosario*, *Usuari*
	Eliminar palabra del glosario	*Administración*, *Gestionar glosario*, *Usuari*
	Cargar entradas de glosario	*Administración*, *Gestionar glosario*, *Usuari*
Sugerencias automáticas	Utilizar sugerencias automáticas	*Administración*, *Editar fuente*, *Usuario ava*
Memoria de traducción	Editar la memoria de traducción	*Administración*, *Gestión de la memoria de*
	Eliminar memoria de traducción	*Administración*, *Gestión de la memoria de*
Proyectos	Editar configuración del proyecto	*Administración*
	Gestionar acceso al proyecto	*Administración*
Informes	Descargar informes	*Administración*
Capturas de pantalla	Añadir captura de pantalla	*Administración*, *Gestión de capturas de pa*
	Editar captura de pantalla	*Administración*, *Gestión de capturas de pa*
	Eliminar captura de pantalla	*Administración*, *Gestión de capturas de pa*
Cadenas de origen	Editar información adicional de cadena	*Administración*, *Editar código*
Cadenas	Añadir nueva cadena	*Administración*
	Quitar una cadena	*Administración*
	Descartar comprobación fallida	*Administración*, *Editar fuente*, *Usuario ava*
	Editar cadenas	*Administración*, *Editar fuente*, *Usuario ava*
	Revisar cadenas	*Administración*, *Revisar cadenas*
	Editar cadena cuando son obligatorias las sugerencias	*Administración*, *Revisar cadenas*
	Editar cadenas de origen	*Administración*, *Editar origen*, *Usuario ava*
Sugerencias	Aceptar sugerencia	*Administración*, *Editar fuente*, *Usuario ava*
	Añadir sugerencia	*Administración*, *Editar fuente*, *Añadir suge*
	Eliminar sugerencia	*Administración*, *Usuario avanzado*
	Votar sugerencia	*Administración*, *Editar fuente*, *Usuario ava*
Traducciones	Añadir idioma para traducción	*Administración*, *Usuario avanzado*, *Admin*
	Realizar una traducción automática	*Administración*, *Gestión de idiomas*
	Eliminar traducción existente	*Administración*, *Gestión de idiomas*
	Descargar archivo de traducción	*Administración*, *Editar fuente*, *Acceder al*
	Añadir varios idiomas para traducción	*Administración*, *Gestión de idiomas*
Cargas	Definir autor/a de la traducción cargada	*Administración*

Tabla 4 – proviene de la página anterior

Alcance	Permiso	Roles
Sistema de control de versiones	Sobrescribir cadenas existentes al subir	*Administración*, *Editar fuente*, *Usuario ava*
	Cargar traducciones	*Administración*, *Editar fuente*, *Usuario ava*
	Acceso al repositorio interno	*Administración*, *Acceso al repositorio*, *Usu*
	Enviar cambios al repositorio interno	*Administración*, *Gestionar repositorio*
	Enviar cambios del repositorio interno	*Administración*, *Gestionar repositorio*
	Deshacer los cambios del repositorio interno	*Administración*, *Gestionar repositorio*
	Ver la ubicación del repositorio originario	*Administración*, *Acceso al repositorio*, *Usu*
	Actualizar el repositorio interno	*Administración*, *Gestionar repositorio*
Privilegios en todo el sitio	Utilizar interfaz de gestión	
	Añadir proyectos nuevos	
	Añadir definiciones de idioma	
	Gestionar definiciones de idioma	
	Dirigir equipos	
	Administrar usuarios	
	Gestionar roles	
	Gestionar anuncios	
	Gestionar memoria de traducción	
	Gestionar automatización	
	Gestionar listas de componentes	

Nota: Los privilegios para todo el sitio no se conceden a ningún rol por defecto. Estos son poderosos y bastante cercanos al estatus de superusuario. La mayoría de ellos afectan a todos los proyectos en tu instalación de Weblate.

Lista de los equipos

The following teams are created upon installation (or after executing `weblate weblate setupgroups`) and you are free to modify them. The migration will, however, re-create them if you delete or rename them.

Invitados
: Define los permisos que tendrán los usuarios no autenticados.

  Este equipo sólo contiene usuarios anónimos (véase `ANONYMOUSER_NAME`).

  Puedes eliminar las funciones de este equipo para limitar los permisos de los usuarios no registrados.

  Roles por defecto: *Añadir sugerencia*, *Acceder al repositorio*

Lectores
: This role ensures visibility of public projects for all users. By default, all users are members of this team.

  By default, *automatic team assignment* makes all new accounts members of this team when they join.

  Roles por defecto: ninguno

Usuario/as
: Default team for all users.

  By default, *automatic team assignment* makes all new accounts members of this team when they join.

  Roles por defecto: *Usuario/a avanzado/a*

Revisores
: Grupo de revisores (ver *Flujos de trabajo de traducción*).

  Roles por defecto: *Cadenas de revisión*

Supervisores
: Grupo para administradores.

Roles por defecto: *Administración*

Advertencia: Never remove the predefined Weblate teams and users as this can lead to unexpected problems! If you have no use for them, you can removing all their privileges instead.

2.6.3 Restricciones de acceso adicionales

Si deseas utilizar tu instalación de Weblate de una manera menos pública, es decir, permitir nuevos usuarios sólo por invitación, puedes hacerlo configurando Weblate de tal manera que sólo los usuarios conocidos tengan acceso a él. Para ello, puedes fijar *REGISTRATION_OPEN* a `False` para evitar el registro de nuevos usuarios, y fijar *REQUIRE_LOGIN* a `/.*` para requerir el inicio de sesión para acceder a todas las páginas del sitio. Esta es básicamente la forma de bloquear tu instalación de Weblate.

Consejo: Puedes usar la función invite-user> para añadir nuevos usuarios.

2.7 Proyectos de traducción

2.7.1 Organización de traducción

Weblate organiza el contenido traducible del sistema de control de versiones del proyecto o los componentes en una estructura semejante a un árbol.

- The bottom level object is *Configuración de proyectos*, which should hold all translations belonging together (for example translation of an application in several versions and/or accompanying documentation).
- On the level above, *Configuración de componentes*, which is actually the component to translate, you define the VCS repository to use, and the mask of files to translate.
- Above *Configuración de componentes* there are individual translations, handled automatically by Weblate as translation files (which match *Máscara de archivos* defined in *Configuración de componentes*) appear in the VCS repository.

Weblate supports a wide range of translation formats (both bilingual and monolingual ones) supported by Translate Toolkit, see *Formatos de archivo admitidos*.

Nota: You can share cloned VCS repositories using *URL internos de Weblate*. Using this feature is highly recommended when you have many components sharing the same VCS. It improves performance and decreases required disk space.

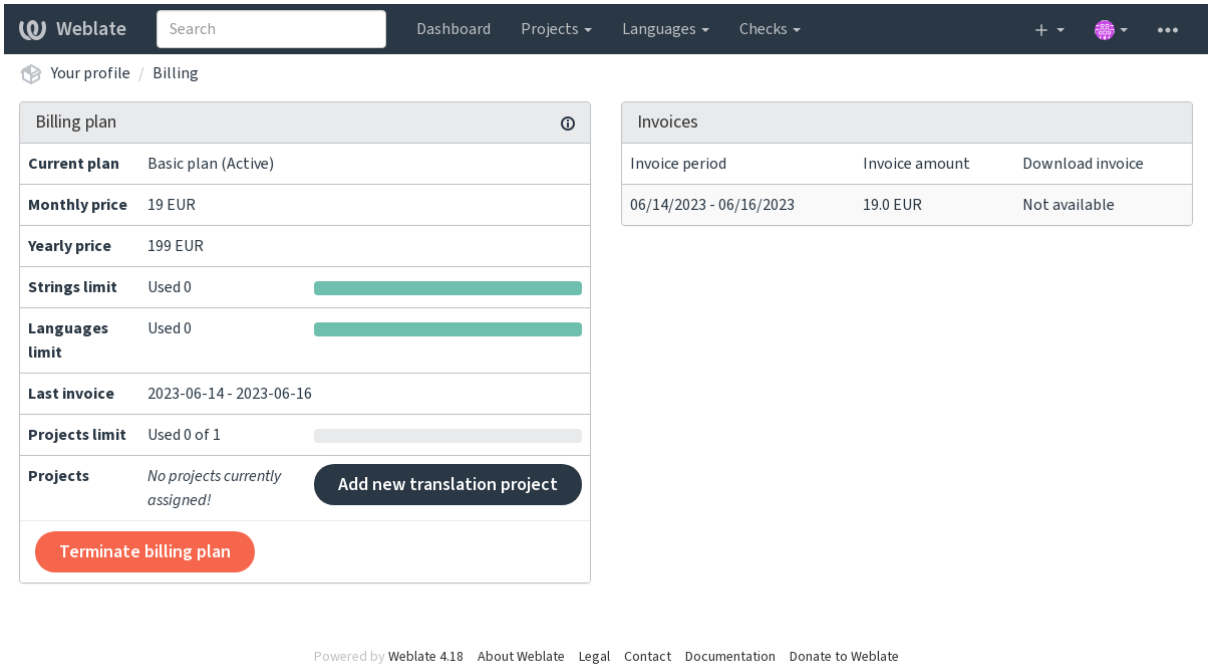
2.7.2 Añadir proyectos y componentes de traducción

Distinto en la versión 3.2: An interface for adding projects and components is included, and you no longer have to use *La interfaz administrativa de Django*.

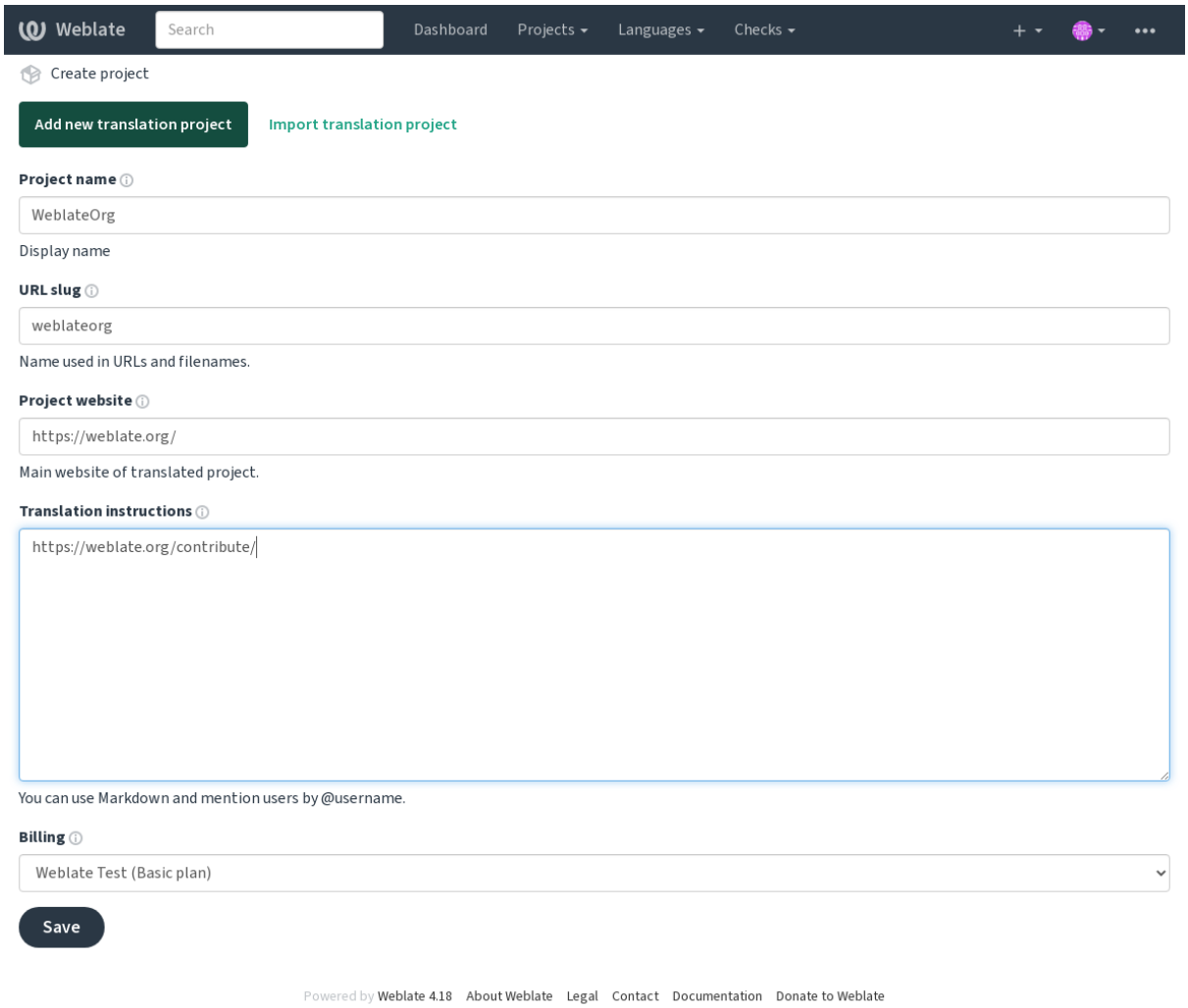
Distinto en la versión 3.4: The process of adding components is now multi staged, with automated discovery of most parameters.

Based on your permissions, new translation projects and components can be created. It is always permitted for users with the *Add new projects* permission, and if your instance uses billing (e.g. like https://hosted.weblate.org/ see *Facturación*), you can also create those based on your plans allowance from the user account that manages billing.

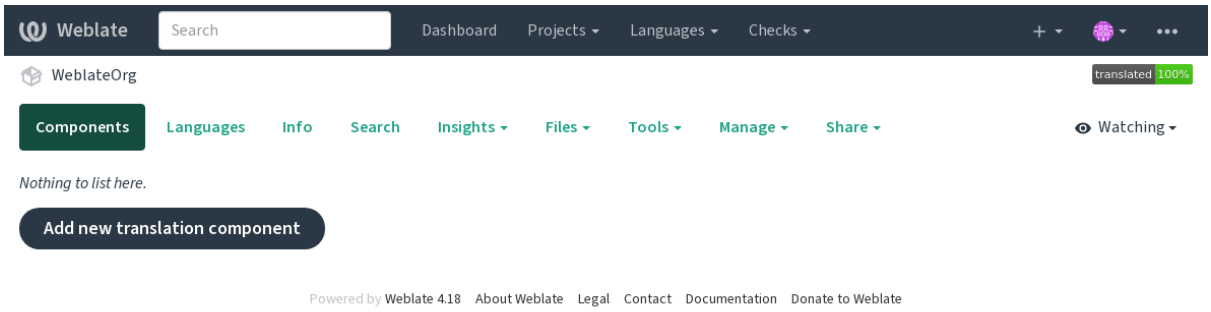
Puede ver su plan de facturación actual en una página separada:



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:



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:

Desde control de versiones
: Crea el componente a partir de un repositorio de control de versiones remoto.

Desde componente existente
: Creates additional component to existing one by choosing different files.

Rama adicional
: Creates additional component to existing one, just for different branch.

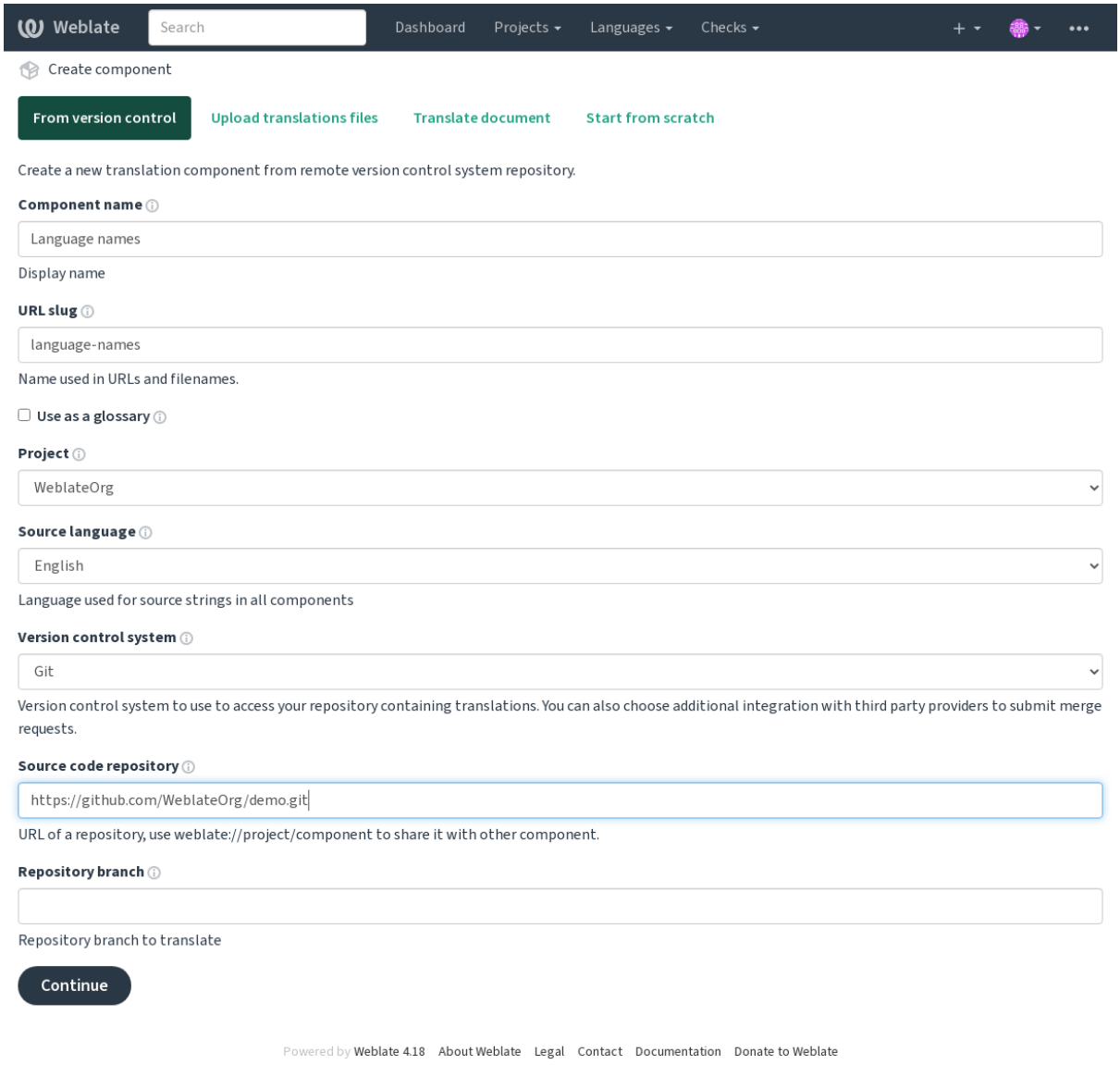
Cargar archivos de traducción
: Upload translation files to Weblate in case you do not have version control or do not want to integrate it with Weblate. You can later update the content using the web interface or *API REST de Weblate*.

Traducir documento
: Suba un solo documento o archivo de traducción y tradúcelo.

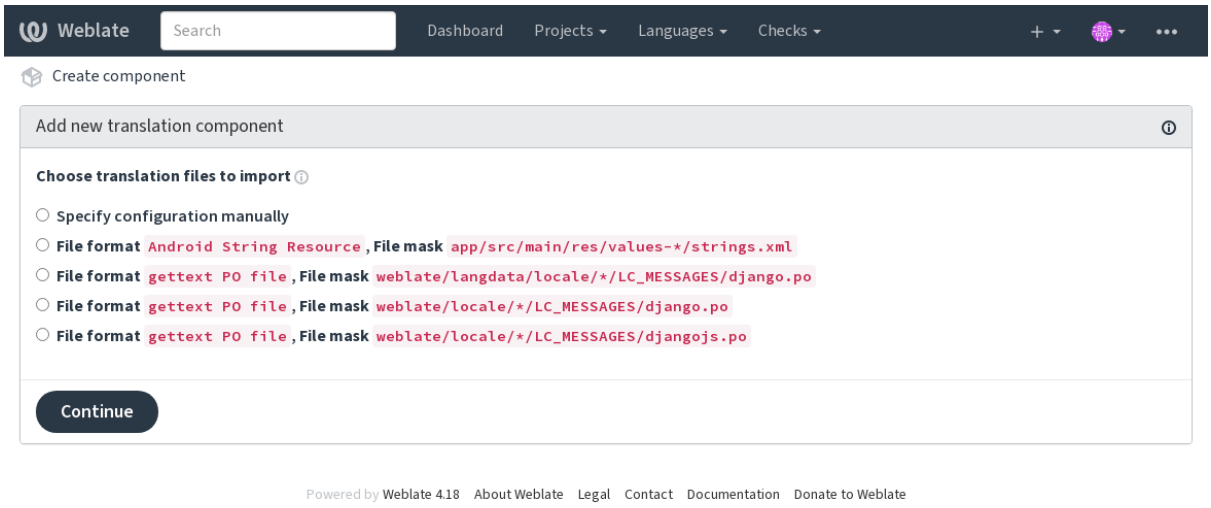
Comenzar de cero
: Create blank translation project and add strings manually.

Once you have existing translation components, you can also easily add new ones for additional files or branches using same repository.

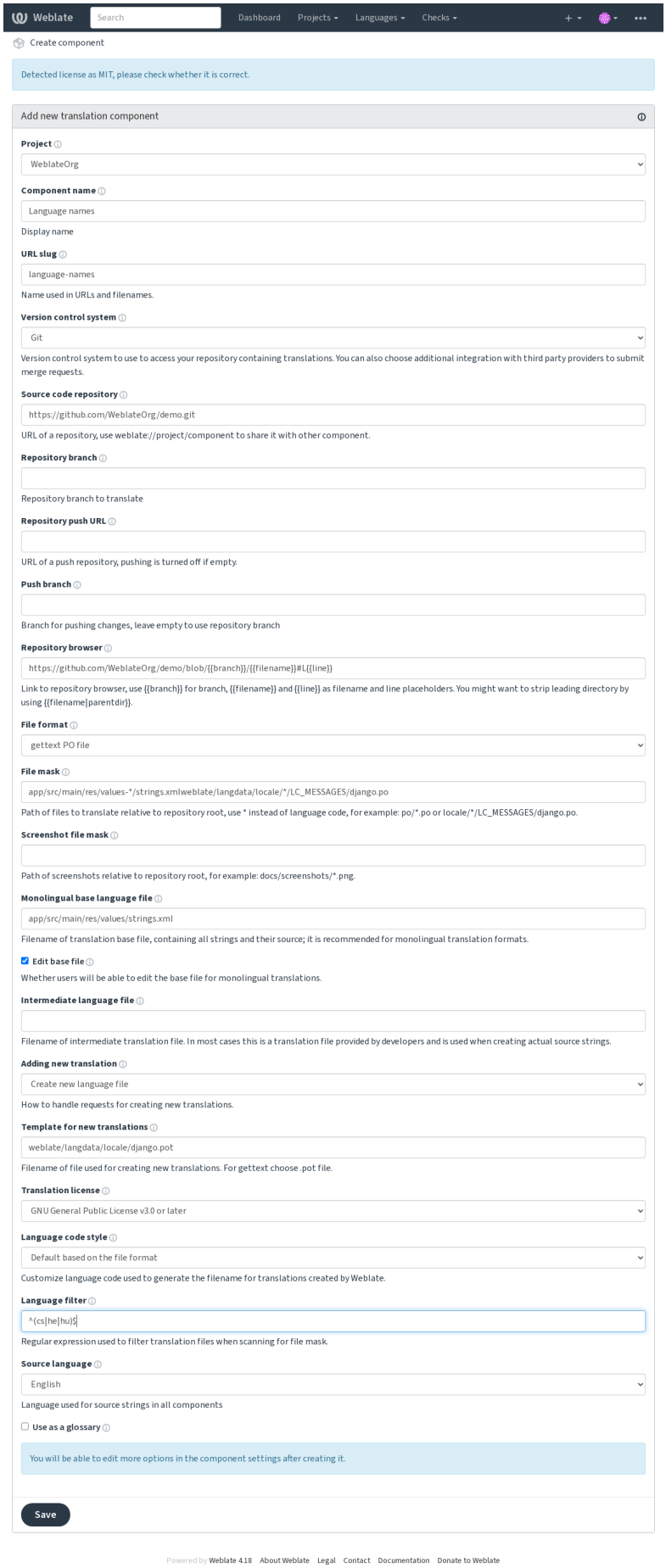
Primero debe suplir el nombre y la ubicación del repositorio:



En la próxima página verá una lista de recursos traducibles detectados:



Como último paso, revise la información del componente de traducción y supla datos opcionales:


Weblate
Search
Dashboard
Projects
Languages
Checks
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.
Screenshot file mask
Path of screenshots relative to repository root, for example: docs/screenshots/*.png.
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
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Ver también:

La interfaz administrativa de Django, *Configuración de proyectos*, *Configuración de componentes*

2.7.3 Configuración de proyectos

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 *Memoria de traducción*.

Ver también:

/devel/integration

These basic attributes set up and inform translators of a project:

Nombre del proyecto

Verbose project name, used to display the project name.

Ver también:

PROJECT_NAME_RESTRICT_RE

«Slug» del URL

Project name suitable for URLs.

Sitio web del proyecto

URL where translators can find more info about the project.

This is a required parameter unless turned off by *WEBSITE_REQUIRED*.

Ver también:

PROJECT_WEB_RESTRICT_HOST, *PROJECT_WEB_RESTRICT_NUMERIC*, *PROJECT_WEB_RESTRICT_RE*

Instrucciones de traducción

Text describing localization process in the project, and any other information useful for translators. Markdown can be used for text formatting or inserting links.

Definir cabecera «Language-Team»

Whether Weblate should manage the `Language-Team` header (this is a *gettext de GNU* only feature right now).

Utilizar memoria de traducción compartida

Whether to use shared translation memory, see *Memoria de traducción compartida* for more details.

El valor predeterminado se puede cambiar `DEFAULT_SHARED_TM`.

Contribuir a la memoria de traducción compartida

Whether to contribute to shared translation memory, see *Memoria de traducción compartida* for more details.

This also affects whether the project can be used as source for *Traducción automática*.

El valor predeterminado se puede cambiar `DEFAULT_SHARED_TM`.

Nota: This option is unavailable on Hosted Weblate, it is toggled together with *Utilizar memoria de traducción compartida*.

Control de acceso

Configure per project access control, see *Control de acceso al proyecto* for more details.

El valor predeterminado se puede cambiar mediante :setting: *DEFAULT_ACCESS_CONTROL*.

Activar revisiones

Enable review workflow for translations, see *Revisores dedicados*.

Activar revisiones de origen

Enable review workflow for source strings, see *Source strings reviews*.

Ver también:

report-source, *Comentarios*

Activar actuadores

Whether unauthenticated *Actuadores de notificación* are to be used for this repository.

Ver también:

Archivo de idioma intermediario, *Quality gateway for the source strings*, *Formatos bilingües y monolingües*, *Definiciones de idioma*

Alias de idiomas

Definir el mapeo de los códigos de idioma al importar las traducciones a Weblate. Utilícelo cuando los códigos del idioma sean inconsistentes en sus repositorios y quiera obtener una vista consistente en Weblate o en caso de que quiera utilizar una nomenclatura no estándar para sus archivos de traducción.

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`

Consejo: 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.

Ver también:

adding-translation, *Código de idioma*, *Procesamiento de códigos de idioma*

2.7.4 Configuración de componentes

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.

Ver también:

/devel/integration

You can find some examples of typical configurations in the *Formatos de archivo admitidos*.

Nota: Se recomienda mantener los componentes de traducción en un tamaño razonable: divida la traducción por cualquier cosa que tenga sentido en su caso (aplicaciones o complementos individuales, capítulos de libros o sitios web).

Weblate puede manejar con facilidad traducciones de decenas de miles de cadenas, pero es más difícil distribuir el trabajo y coordinar los traductores al utilizar componentes de tales dimensiones.

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:

Nombre de componente

Verbose component name, used to display the component name.

«Slug» del componente

Component name suitable for URLs.

Proyecto del componente

Configuración de proyectos where the component belongs.

Sistema de control de versiones

VCS to use, see *Integración de control de versiones* for details.

Ver también:

Enviar cambios efectuados en Weblate

Repositorio de código fuente

VCS repository used to pull changes.

Ver también:

See *Accessing repositories* for more details on specifying URLs.

Consejo: This can either be a real VCS URL or `weblate://project/component` indicating that the repository should be shared with another component. See *URL internos de Weblate* for more details.

URL de envío al repositorio

Repository URL used for pushing. The behavior of this depends on *Sistema de control de versiones*, and this is in more detail covered in *Enviar cambios efectuados en Weblate*.

For linked repositories, this is not used and setting from linked component applies.

Ver también:

See *Accessing repositories* for more details on how to specify a repository URL and *Enviar cambios efectuados en Weblate* for more details on pushing changes from Weblate.

Explorador del repositorio

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use *Marcación de plantilla*.

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 *Marcación de plantilla*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

URL de repositorio exportado

URL where changes made by Weblate are exported. This is important when *Regionalización continua* is not used, or when there is a need to manually merge changes. You can use *Git exporter* to automate this for Git repositories.

Rama del repositorio

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.

Rama a la que enviar

Branch for pushing changes, leave empty to use *Rama del repositorio*.

For linked repositories, this is not used and setting from linked component applies.

Nota: This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

Ver también:

Enviar cambios efectuados en Weblate

Máscara de archivos

Mask of files to translate, including path. It should include one «*» replacing language code (see *Definiciones de idioma* 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 `[]]`.

Ver también:

Formatos bilingües y monolingües, *What does mean «There are more files for the single language (en)»?*

Screenshot file mask

Mask of files to allow discovery/update of screenshots, including path from vcs repository. It should include one «*» replacing screenshot file name.

Allowed formats are JPEG, PNG, APNG and GIF.

For example `docs/*.png`.

Archivo de base monolingüe

Base file containing string definitions for *Componentes monolingües*.

Ver también:

Formatos bilingües y monolingües, *What does mean «There are more files for the single language (en)»?*

Editar archivo de base

Whether to allow editing the base file for *Componentes monolingües*.

Archivo de idioma intermediario

Intermediate language file for *Componentes monolingües*. 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 *Archivo de base monolingüe*. In case the string is not translated into the source language, translating to other languages is prohibited. This provides *Quality gateway for the source strings*.

Ver también:

Quality gateway for the source strings, *Formatos bilingües y monolingües*, *What does mean «There are more files for the single language (en)»?*

Plantilla para traducciones nuevas

Base file used to generate new translations.

- Keep this field empty for most of the monoligual formats. Those are typically able to start from an empty file.
- Choose `.pot` file with GNU gettext PO files.
- Choose blank file without translations, if you have one.
- Choose *Archivo de base monolingüe* for monolingual formats that need a full set of keys present.
- Choose *Archivo de base monolingüe* for document translations.

- Choose any translation file for others.
- Template file can be the same as the base file in most cases.

Consejo: 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.

Ver también:

adding-translation, *Adición de traducciones nuevas*, *Formatos bilingües y monolingües*, *What does mean «There are more files for the single language (en)»?*

Formato de archivo

Translation file format, see also *Formatos de archivo admitidos*.

Dirección para informar de errores en las cadenas de origen

Email address used for reporting upstream bugs. This address will also receive notification about any source string comments made in Weblate.

Permitir propagación de traducciones

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`.

Ver también:

Mantener iguales las traducciones entre los componentes

Activar sugerencias

Whether translation suggestions are accepted for this component.

Votar sugerencias

Turns on vote casting for suggestions, see *Votar sugerencias*.

Aceptar sugerencias automáticamente

Automatically accept voted suggestions, see *Votar sugerencias*.

Indicadores de traducción

Customization of quality checks and other Weblate behavior, see *Personalizar el comportamiento mediante indicadores*.

Comprobaciones obligatorias

List of checks which can not be ignored, see *Forzar comprobaciones*.

Nota: Enforcing the check does not automatically enable it, you still should enabled it using *Personalizar el comportamiento mediante indicadores* in *Indicadores de traducción* or *Información adicional sobre las cadenas de origen*.

Licencia de la traducción

License of the translation (does not need to be the same as the source code license).

Acuerdo de contribuidor

Acuerdo de usuario que debe aprobarse antes de que un usuario pueda traducir este componente.

Adición de traducciones nuevas

How to handle requests for creation of new languages. Available options:

Contactar a responsables
: 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.

Apuntar al URL con instrucciones de traducción
: 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).

Crear archivo de idioma nuevo
: User can select language and Weblate automatically creates the file for it and translation can begin.

Desactivar adición de traducciones nuevas
: There will be no option for user to start new translation.

Consejo: The project admins can add new translations even if it is disabled here when it is possible (either *Plantilla para traducciones nuevas* or the file format supports starting from an empty file).

Ver también:

adding-translation, *Plantilla para traducciones nuevas*

Gestionar cadenas

Nuevo en la versión 4.5.

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.

Ver también:

Formatos bilingües y monolingües, adding-new-strings, *POST /api/translations/(string:project)/(string:component)/(string:language)/units/*

Estilo de código de idioma

Personalice el código de idioma utilizado para generar el nombre de archivo de las traducciones creadas por Weblate.

Nota: Weblate recognizes any of the language codes when parsing translation files, following settings only influences how new files are created.

Basado en el formato de archivo predefinido
Depende del formato de archivo; en la mayoría se utiliza POSIX.

Estilo POSIX con guion bajo como separador
Typically used by gettext and related tools, produces language codes like `pt_BR`.

Estilo POSIX con guion bajo como separador; incluye el código de país
POSIX style language code including the country code even when not necessary (for example `cs_CZ`).

POSIX style using underscore as a separator, including country code (lowercase)
POSIX style language code including the country code even when not necessary (lowercase) (for example `cs_cz`).

Estilo BCP con guion como separador
Typically used on web platforms, produces language codes like `pt-BR`.

Estilo BCP con guion como separador; incluye el código de país
BCP style language code including the country code even when not necessary (for example `cs-CZ`).

Estilo BCP que utiliza el guión como separador, códigos de idiomas heredados
Uses legacy codes for Chinese and BCP style notation.

Estilo BCP con guión como separador, en minúsculas
BCP style notation, all in lower case (for example `cs-cz`).

Estilo a los metadatos de la App Store de Apple
Style suitable for uploading metadata to Apple App Store.

Estilo de los metadatos de Google Play
Style suitable for uploading metadata to Google Play Store.

Estilo Android
Only used in Android apps, produces language codes like `pt-rBR`.

Estilo Linux
Locales as used by Linux, uses legacy codes for Chinese and POSIX style notation.

Ver también:

adding-translation, *Código de idioma*, *Procesamiento de códigos de idioma*, *Alias de idiomas*

Estilo de fusión

You can configure how updates from the upstream repository are handled. The actual implementation depends on VCS, see *Integración de control de versiones*.

Cambiar base («rebase»)
: Rebases Weblate commits on top of upstream repository on update. This provides clean history without extra merge commits.

  Rebasing can cause you trouble in case of complicated merges, so carefully consider whether or not you want to enable them.

  You might need to enable force pushing by choosing *Git con envío forzado* as *Sistema de control de versiones*, especially when pushing to a different branch.

Fusionar
: 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.

Fusionar sin avance rápido
: 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`.

Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud

Message used when committing a translation, see *Marcación de plantilla*.

El valor predeterminado se puede cambiar **:configurando:`DEFAULT_ADD_MESSAGE`**, **:configurando:`DEFAULT_ADDON_MESSAGE`**, **:configurando:`DEFAULT_COMMIT_MESSAGE`**, **:configurando:`DEFAULT_DELETE_MESSAGE`**, **:configurando:`DEFAULT_MERGE_MESSAGE`**, **:configurando:`DEFAULT_PULL_MESSAGE`**.

Enviar al consignar

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 *Consignas diferidas*). To actually enable pushing *Repository push URL* has to be configured as well.

Antigüedad de cambios por consignar

Sets how old (in hours) changes have to be before they are committed by background task or the `weblate weblate commit_pending` management command. All changes in a component are committed once there is at least one change older than this period.

Default value can be changed by `COMMIT_PENDING_HOURS`.

Consejo: There are other situations where pending changes might be committed, see *Consignas diferidas*.

Bloquear al producirse un error

Locks the component (and linked components, see *URL internos de Weblate*) upon the first failed push or merge into its upstream repository, or pull from it. This avoids adding another conflicts, which would have to be resolved manually.

The component will be automatically unlocked once there are no repository errors left.

Idioma del código fuente

Language used for source strings. Change this if you are translating from something else than English.

Consejo: 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 *Archivo de idioma intermediario*.

Filtro de idioma

Expresión regular utilizada para filtrar la traducción al buscar una máscara del archivo. Se puede utilizar para limitar la lista de idiomas gestionados por Weblate.

Nota: You need to list language codes as they appear in the filename.

Some examples of filtering:

Descripción del filtro	Expresión regular
Solo los idiomas seleccionados	`^(cs|de|es)$`
Excluir idiomas	`^(?!(it|fr)$).+$`
Filter two letter codes only	`^..$`
Excluir archivos no lingüísticos	`^(?!(blank)$).+$`
Incluir todos los archivos (predeterminado)	`^[^.]+$`

Expresión regular de variantes

Regular expression used to determine the variants of a string, see variants.

Nota: La mayoría de los campos pueden ser editados por los propietarios o administradores del proyecto en la interfaz de Weblate.

Ver también:

¿Weblate admite sistemas de control de versiones aparte de Git y Mercurial?, alerts

Prioridad

Se ofrecen primero a los traductores los componentes con mayor prioridad.

Distinto en la versión 4.15: This now also affects ordering of matched glossary terms.

Acceso restringido

Nota: Esta función no está disponible en Hosted Weblate.

By default the component is visible to anybody who has access to the project, even if the person can not perform any changes in the component. This makes it easier to keep translation consistency within the project.

Restricting access at a component, or component-list level takes over access permission to a component, regardless of project-level permissions. You will have to grant access to it explicitly. This can be done through granting access to a new user group and putting users in it, or using the default *custom* or *private* access control groups.

The default value can be changed in `DEFAULT_RESTRICTED_COMPONENT`.

Consejo: This applies to project admins as well — please make sure you will not loose access to the component after toggling the status.

Compartir en proyectos

Puede elegir proyectos adicionales donde el componente será visible. Útil para bibliotecas compartidas para utilizar en varios proyectos.

Nota: Compartir un componente no cambia su control de acceso. Solo lo hace visible al navegar por otros proyectos. Los usuarios aún necesitan acceso al componente real para examinarlo o traducirlo.

Utilizar como glosario

Nuevo en la versión 4.5.

Permite la utilización de este componente como glosario. Puede definir cómo se mostrará en las listas mediante *Color de glosario*.

El glosario será accesible en todos los proyectos definidos por *Compartir en proyectos*.

Se recomienda la activación de *Gestionar cadenas* en los glosarios para permitir la adición de palabras nuevas.

Ver también:

Glosario

Color de glosario

Display color for a glossary used when showing word matches.

2.7.5 Marcación de plantilla

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.

Actualmente se utiliza en:

- Commit message formatting, see *Configuración de componentes*
- **Varios complementos**
  - *Detección de componentes*
  - *Generador de estadísticas*
  - *Ejecutar scripts desde el complemento*

Las variables siguientes están disponibles en las plantillas de componente:

{{ language_code }}
Código de idioma

{{ language_name }}
Nombre del idioma

{{ component_name }}
Nombre de componente

{{ component_slug }}
«Slug» del componente

{{ project_name }}
Nombre del proyecto

{{ project_slug }}
«Slug» del proyecto

{{ url }}
URL de traducción

{{ filename }}
Nombre de archivo de traducción

{{ stats }}
Translation stats, this has further attributes, examples below.

{{ stats.all }}
Total strings count

{{ stats.fuzzy }}
Count of strings needing review

{{ stats.fuzzy_percent }}
Percent of strings needing review

{{ stats.translated }}
Translated strings count

{{ stats.translated_percent }}
Translated strings percent

{{ stats.allchecks }}
Number of strings with failing checks

{{ stats.allchecks_percent }}
Percent of strings with failing checks

{{ author }}
Author of current commit, available only in the commit scope.

{{ addon_name }}
Name of currently executed add-on, available only in the add-on commit message.

The following variables are available in the repository browser or editor templates:

{{branch}}
current branch

{{line}}
line in file

{{filename}}
filename, you can also strip leading parts using the `parentdir` filter, for example `{{filename|parentdir}}`

Consejo: In some places additional variables can be available, see *Detección de componentes*.

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|stripext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times:  {{ filename|parentdir|parentdir }}
```

…and other Django template features.

2.7.6 Importing speed

Fetching VCS repository and importing translations to Weblate can be a lengthy process, depending on size of your translations. Here are some tips:

Optimize configuration

The default configuration is useful for testing and debugging Weblate, while for a production setup, you should do some adjustments. Many of them have quite a big impact on performance. Please check *Puesta en marcha de entorno de producción* for more details, especially:

- Configure Celery for executing background tasks (see *Tareas en segundo plano con Celery*)
- *Enable caching*
- *Use a powerful database engine*
- *Desactivar el modo de depuración*

Check resource limits

If you are importing huge translations or repositories, you might be hit by resource limitations of your server.

- Check the amount of free memory, having translation files cached by the operating system will greatly improve performance.
- Disk operations might be bottleneck if there is a lot of strings to process—the disk is pushed by both Weblate and the database.
- Additional CPU cores might help improve performance of background tasks (see *Tareas en segundo plano con Celery*).

Disable unneeded checks

Some quality checks can be quite expensive, and if not needed, can save you some time during import if omitted. See `CHECK_LIST` for info on configuration.

2.7.7 Creación automática de componentes

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 `weblate weblate import_project` or `weblate weblate import_json`, or by installing the *Detección de componentes* 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 `weblate weblate import_project` or `weblate weblate import_json`.

Ver también:

Órdenes de gestión, *Detección de componentes*

2.8 Definiciones de idioma

Para presentar adecuadamente cada una de las traducciones, es necesario contar con información sobre el nombre del idioma, la dirección del texto, la definición de los plurales y el código de lengua.

2.8.1 Definiciones de idioma incorporadas

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 weblate migrate` is executed, see *Instrucciones de actualización genéricas*) the database of languages is updated to include all language definitions shipped in Weblate.

This feature can be disable using `UPDATE_LANGUAGES`. You can also enforce updating the database to match Weblate built-in data using `weblate weblate setuplang`.

Ver también:

Extending built-in language definitions, Current language definitions

2.8.2 Procesamiento de códigos de idioma

While parsing translations, Weblate attempts to map language code (usually the ISO 639-1 one) from the *Máscara de archivos* to any existing language object.

You can further adjust this mapping at project level by *Alias de idiomas*.

If no exact match can be found, an attempt will be made to best fit it into an existing language. Following steps are tried:

- Búsquedas sin distinción de mayúsculas y minúsculas.
- Normalización de guiones bajos y guiones.
- Búsqueda de alias de idiomas incorporados.
- Búsqueda por nombre de idioma.
- Ignoring the default country code for a given language—choosing `cs` instead of `cs_CZ`.

Should that also fail, a new language definition will be created using the defaults (left to right text direction, one plural). The automatically created language with code `xx_XX` will be named as *xx_XX (generated)*. You might want to change this in the admin interface later, (see *Cambiar las definiciones de idioma*) and report it to the issue tracker (see *Contribuir con Weblate*), so that the proper definition can be added to the upcoming Weblate release.

Consejo: In case you see something unwanted as a language, you might want to adjust *Filtro de idioma* to ignore such file when parsing translations.

Ver también:

Código de idioma, adding-translation

2.8.3 Cambiar las definiciones de idioma

Es posible cambiar las definiciones de idioma en la interfaz de idiomas (URL `/languages/`).

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 Códigos de lenguaje ambiguos y macrolenguajes

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.

Ver también:

Macrolanguages definition, List of macrolanguages

2.8.5 Definiciones de idioma

Each language consists of following fields:

Código de idioma

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.

Ver también:

Procesamiento de códigos de idioma, adding-translation

Nombre del idioma

Visible name of the language. The language names included in Weblate are also being localized depending on user interface language.

Dirección del texto

Determines whether language is written right to left or left to right. This property is autodetected correctly for most of the languages.

Plural number

Number of plurals used in the language.

Fórmula de plurales

Gettext compatible plural formula used to determine which plural form is used for given count.

Ver también:

Plurales, GNU gettext utilities: Plural forms, Language Plural Rules by the Unicode Consortium

Número de hablantes

Number of worldwide speakers of this language.

2.9 Regionalización continua

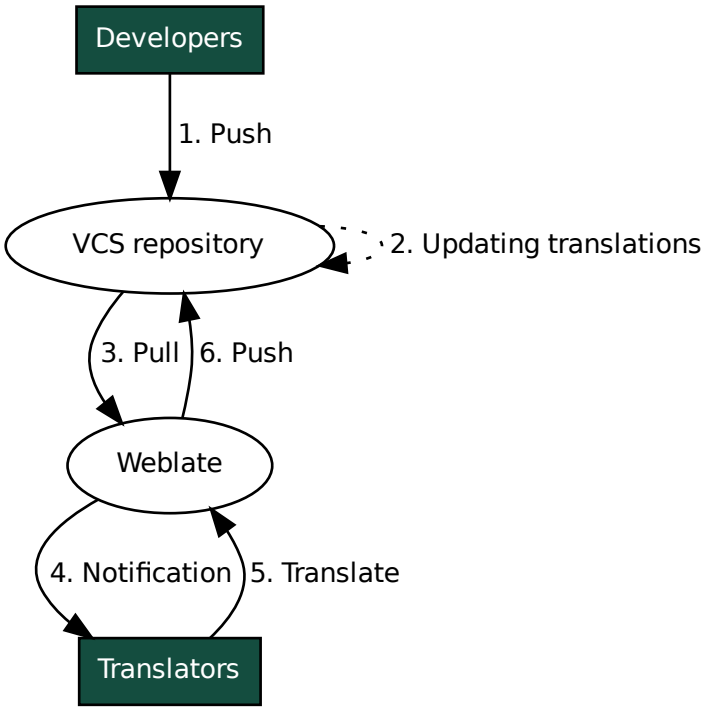
There is infrastructure in place so that your translation closely follows development. This way translators can work on translations the entire time, instead of working through huge amount of new text just prior to release.

Ver también:

/devel/integration describes basic ways to integrate your development with Weblate.

This is the process:

1. Developers make changes and push them to the VCS repository.
2. Optionally the translation files are updated, see translations-update.
3. Weblate pulls changes from the VCS repository, see *Updating repositories*.
4. Once Weblate detects changes in translations, translators are notified based on their subscription settings.
5. Translators submit translations using the Weblate web interface, or upload offline changes.
6. Once the translators are finished, Weblate commits the changes to the local repository (see *Consignas diferidas*) and pushes them back if it has permissions to do so (see *Enviar cambios efectuados en Weblate*).



2.9.1 Updating repositories

You should set up some way of updating backend repositories from their source.

- Use *Actuadores de notificación* to integrate with most of common code hosting services:
  - *Recibir cambios automáticamente de GitHub*
  - *Automatically receiving changes from GitLab*
  - *Recibir cambios automáticamente de Bitbucket*
  - *Recibir cambios automáticamente de Pagure*
  - *Recibir cambios automáticamente de Azure Repos*
  - *Recibir cambios automáticamente de Gitea*
- Manually trigger update either in the repository management or using *API REST de Weblate* or *Cliente de Weblate*
- Enable `AUTO_UPDATE` to automatically update all components on your Weblate instance
- Execute `weblate weblate updategit` (with selection of project or `--all` to update all)

Whenever Weblate updates the repository, the post-update addons will be triggered, see *Complementos*.

Evitar conflictos de fusión

The merge conflicts from Weblate arise when same file was changed both in Weblate and outside it. There are two approaches to deal with that - avoid edits outside Weblate or integrate Weblate into your updating process, so that it flushes changes prior to updating the files outside Weblate.

The first approach is easy with monolingual files - you can add new strings within Weblate and leave whole editing of the files there. For bilingual files, there is usually some kind of message extraction process to generate translatable files from the source code. In some cases this can be split into two parts - one for the extraction generates template (for example gettext POT is generated using **`xgettext`**) and then further process merges it into actual translations (the gettext PO files are updated using **`msgmerge`**). You can perform the second step within Weblate and it will make sure that all pending changes are included prior to this operation.

The second approach can be achieved by using *API REST de Weblate* to force Weblate to push all pending changes and lock the translation while you are doing changes on your side.

The script for doing updates can look like this:

```
# 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
```

If you have multiple components sharing same repository, you need to lock them all separately:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Nota: The example uses *Cliente de Weblate*, which needs configuration (API keys) to be able to control Weblate remotely. You can also achieve this using any HTTP client instead of wlc, e.g. curl, see *API REST de Weblate*.

Ver también:

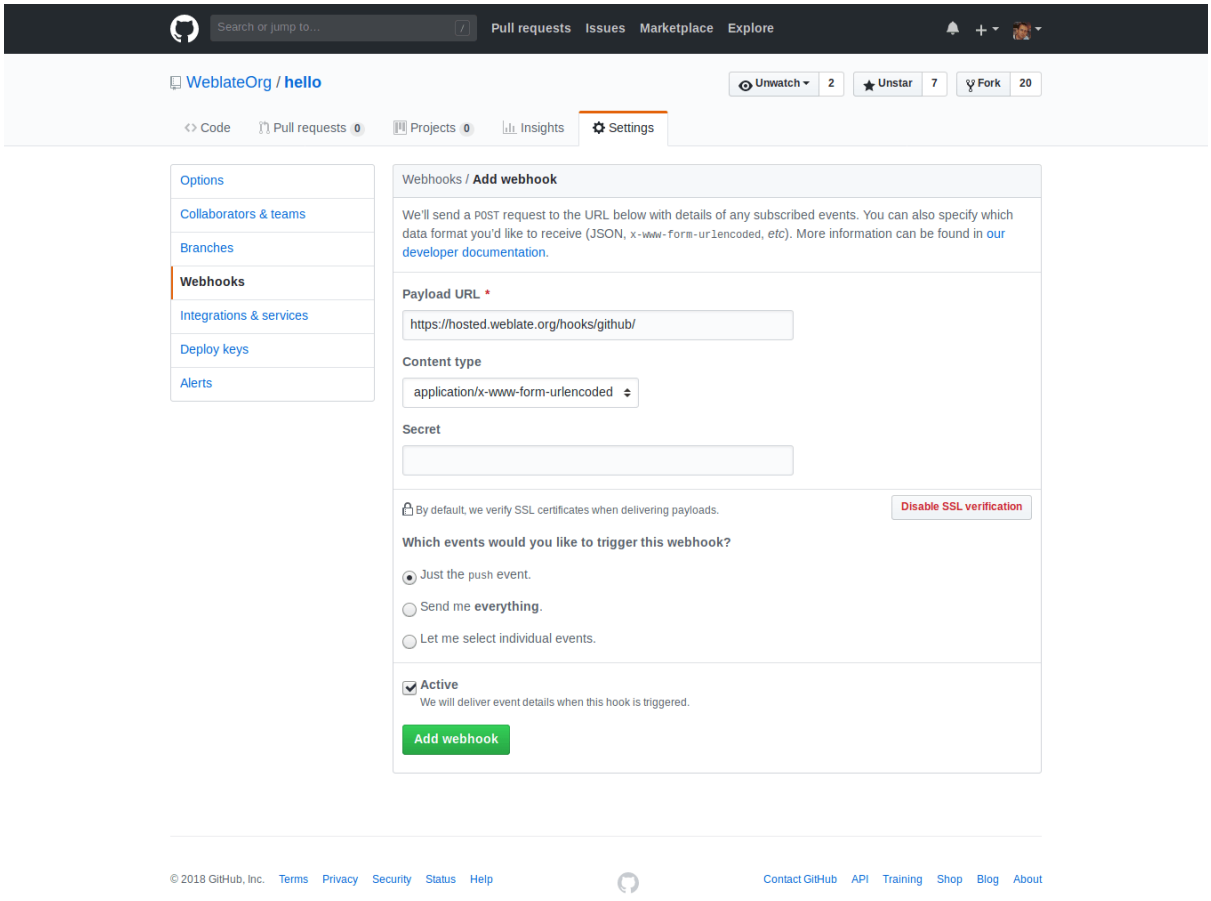
Cliente de Weblate

Recibir cambios automáticamente de GitHub

Weblate admite GitHub nativamente.

If you are using Hosted Weblate, the recommended approach is to install the Weblate app, that way you will get the correct setup without having to set much up. It can also be used for pushing changes back.

To receive notifications on every push to a GitHub repository, add the Weblate Webhook in the repository settings (*Webhooks*) as shown on the image below:



For the payload URL, append `/hooks/github/` to your Weblate URL, for example for the Hosted Weblate service, this is `https://hosted.weblate.org/hooks/github/`.

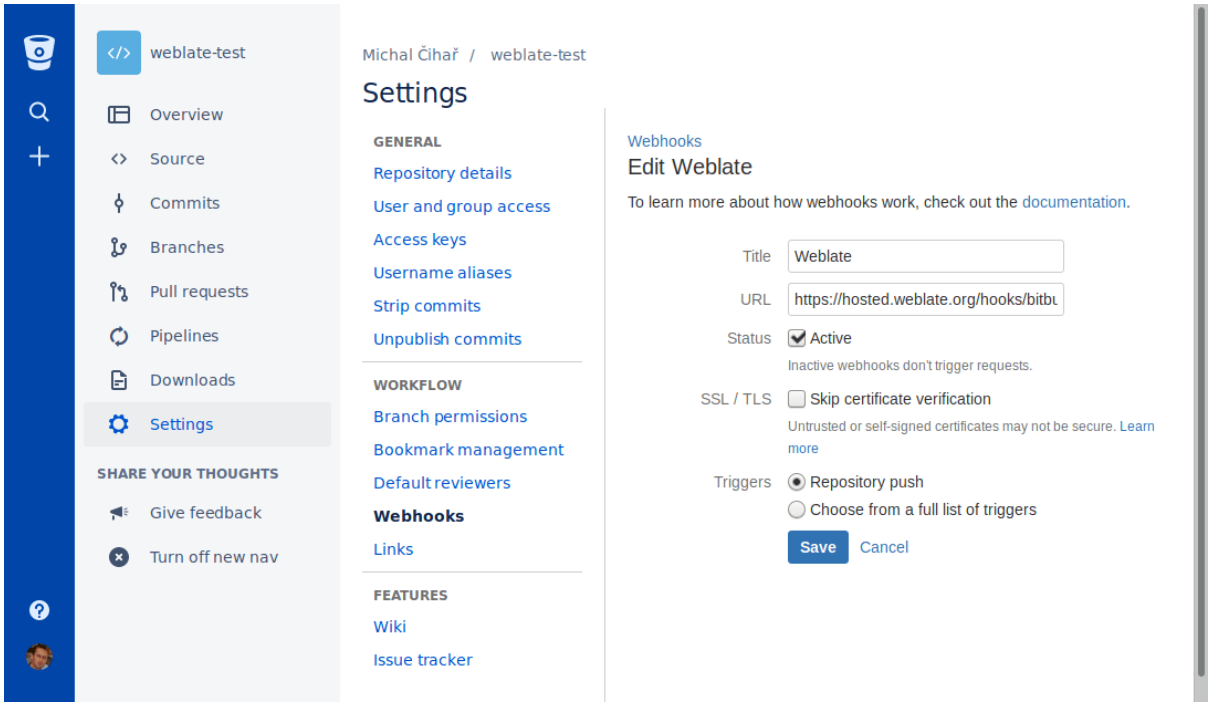
You can leave other values at default settings (Weblate can handle both content types and consumes just the *push* event).

Ver también:

POST /hooks/github/, *Accessing repositories from Hosted Weblate*

Recibir cambios automáticamente de Bitbucket

Weblate has support for Bitbucket webhooks, add a webhook which triggers upon repository push, with destination to `/hooks/bitbucket/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/bitbucket/`).



Ver también:

POST /hooks/bitbucket/, *Accessing repositories from Hosted Weblate*

Automatically receiving changes from GitLab

Weblate has support for GitLab hooks, add a project webhook with destination to `/hooks/gitlab/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitlab/`).

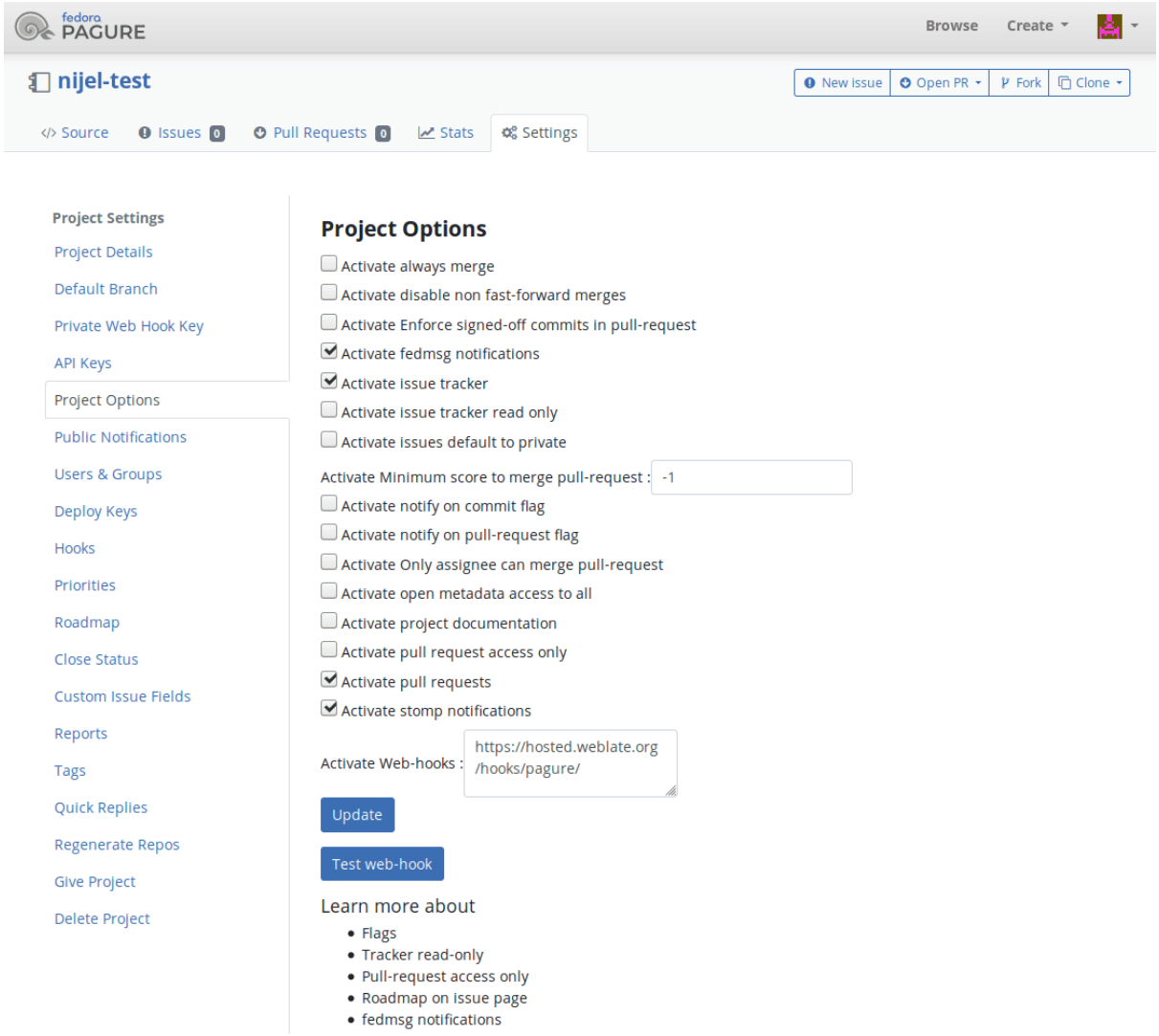
Ver también:

POST /hooks/gitlab/, *Accessing repositories from Hosted Weblate*

Recibir cambios automáticamente de Pagure

Nuevo en la versión 3.3.

Weblate has support for Pagure hooks, add a webhook with destination to `/hooks/pagure/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/pagure/`). This can be done in *Activate Web-hooks* under *Project options*:



Ver también:

`POST /hooks/pagure/`, *Accessing repositories from Hosted Weblate*

Recibir cambios automáticamente de Azure Repos

Nuevo en la versión 3.8.

Weblate has support for Azure Repos web hooks, add a webhook for *Code pushed* event with destination to `/hooks/azure/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/azure/`). This can be done in *Service hooks* under *Project settings*.

Ver también:

Web hooks in Azure DevOps manual, `POST /hooks/azure/`, *Accessing repositories from Hosted Weblate*

Recibir cambios automáticamente de Gitea

Nuevo en la versión 3.9.

Weblate has support for Gitea webhooks, add a *Gitea Webhook* for *Push events* event with destination to `/hooks/gitea/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitea/`). This can be done in *Webhooks* under repository *Settings*.

Ver también:

Webhooks in Gitea manual, `POST /hooks/gitea/`, *Accessing repositories from Hosted Weblate*

Recibir cambios automáticamente de Gitee

Nuevo en la versión 3.9.

Weblate has support for Gitee webhooks, add a *WebHook* for *Push* event with destination to `/hooks/gitee/` URL on your Weblate installation (for example `https://hosted.weblate.org/hooks/gitee/`). This can be done in *WebHooks* under repository *Management*.

Ver también:

Webhooks in Gitee manual, `POST /hooks/gitee/`, *Accessing repositories from Hosted Weblate*

Automatically updating repositories nightly

Weblate automatically fetches remote repositories nightly to improve performance when merging changes later. You can optionally turn this into doing nightly merges as well, by enabling `AUTO_UPDATE`.

2.9.2 Enviar cambios efectuados en Weblate

Each translation component can have a push URL set up (see *URL de envío al repositorio*), and in that case Weblate will be able to push change to the remote repository. Weblate can be also be configured to automatically push changes on every commit (this is default, see *Enviar al consignar*). If you do not want changes to be pushed automatically, you can do that manually under *Repository maintenance* or using API via `wlc push`.

The push options differ based on the *Integración de control de versiones* used, more details are found in that chapter.

In case you do not want direct pushes by Weblate, there is support for *Solicitudes de incorporación de GitHub*, *Solicitudes de fusión de GitLab*, *Solicitudes de incorporación de Gitea*, *Solicitudes de fusión de Pagure* pull requests or *Gerrit* reviews, you can activate these by choosing *GitHub*, *GitLab*, *Gitea*, *Gerrit* or *Pagure* as *Sistema de control de versiones* in *Configuración de componentes*.

De manera general, las opciones siguientes se encuentran disponibles con Git, GitHub y GitLab:

Configuración deseada	*Sistema de control de versiones*	*URL de envío al repositorio*	*Rama a la que enviar*
No push	*Git*	*empty*	*empty*
Enviar directamente	*Git*	URL SSH	*empty*
Enviar en una rama separada	*Git*	URL SSH	Nombre de la rama
No push	*Mercurial*	*empty*	*empty*
Enviar directamente	*Mercurial*	URL SSH	*empty*
Enviar en una rama separada	*Mercurial*	URL SSH	Nombre de la rama
GitHub pull request from fork	*Solicitudes de incorporación de GitHub*	*empty*	*empty*
GitHub pull request from branch	*Solicitudes de incorporación de GitHub*	SSH URL[1]	Nombre de la rama
GitLab merge request from fork	*Solicitudes de fusión de GitLab*	*empty*	*empty*
GitLab merge request from branch	*Solicitudes de fusión de GitLab*	SSH URL[Página 305, 1]	Nombre de la rama
Solicitud de fusión de Gitea desde la bifurcación	*Solicitudes de incorporación de Gitea*	*empty*	*empty*
Gitea la solicitud de fusión desde la rama	*Solicitudes de incorporación de Gitea*	SSH URL[1]	Nombre de la rama
Pagure merge request from fork	*Solicitudes de fusión de Pagure*	*empty*	*empty*
Pagure merge request from branch	*Solicitudes de fusión de Pagure*	SSH URL[1]	Nombre de la rama

Nota: You can also enable automatic pushing of changes after Weblate commits, this can be done in *Enviar al consignar*.

Ver también:

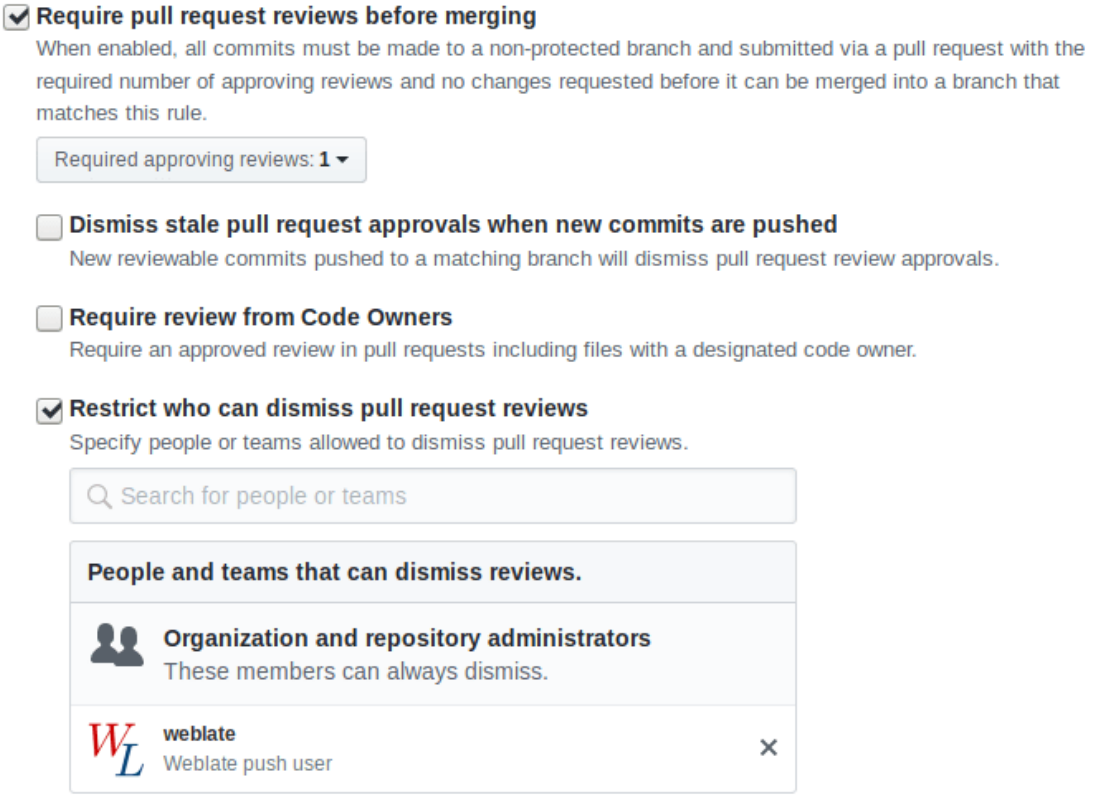
See *Accessing repositories* for setting up SSH keys, and *Consignas diferidas* for info about when Weblate decides to commit changes.

Ramas protegidas

If you are using Weblate on protected branch, you can configure it to use pull requests and perform actual review on the translations (what might be problematic for languages you do not know). An alternative approach is to waive this limitation for the Weblate push user.

For example on GitHub this can be done in the repository configuration:

[1] Can be empty in case *Repositorio de código fuente* supports pushing.



2.9.3 Interactuar con otros

Weblate facilita la interacción con otras herramientas mediante su API.

Ver también:

API REST de Weblate

2.9.4 Consignas diferidas

The behaviour of Weblate is to group commits from the same author into one commit if possible. This greatly reduces the number of commits, however you might need to explicitly tell it to do the commits in case you want to get the VCS repository in sync, e.g. for merge (this is by default allowed for the *Managers* group, see *Lista de privilegios y roles incorporados*).

Los cambios en esta modalidad se consignan una vez que cualquiera de estas condiciones se cumpla:

- Alguien más modifica una cadena ya modificada.
- Se produce una fusión desde el origen ascendente.
- Se solicita explícitamente una consigna.
- Se solicita una descarga de archivos.
- Change is older than period defined as *Antigüedad de cambios por consignar* on *Configuración de componentes*.

Consejo: Commits are created for every component. So in case you have many components you will still see lot of commits. You might utilize *Concentrar consignas de Git* add-on in that case.

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 *La interfaz administrativa de Django*. 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 Procesar el repositorio con secuencias

The way to customize how Weblate interacts with the repository is *Complementos*. Consult *Ejecutar scripts desde el complemento* for info on how to execute external scripts through add-ons.

2.9.6 Mantener iguales las traducciones entre los componentes

Once you have multiple translation components, you might want to ensure that the same strings have same translation. This can be achieved at several levels.

Propagación de traducciones

With *Permitir propagación de traducciones* enabled (what is the default, see *Configuración de componentes*), all new translations are automatically done in all components with matching strings. Such translations are properly credited to currently translating user in all components.

Nota: The translation propagation requires the key to be match for monolingual translation formats, so keep that in mind when creating translation keys.

Comprobación de coherencia

The *Incoherente* check fires whenever the strings are different. You can utilize this to review such differences manually and choose the right translation.

Traducción automática

Automatic translation based on different components can be way to synchronize the translations across components. You can either trigger it manually (see *Traducción automática*) or make it run automatically on repository update using add-on (see *Traducción automática*).

2.10 Licensing translations

You can specify which license translations are contributed under. This is especially important to do if translations are open to the public, to stipulate what they can be used for.

You should specify *Configuración de componentes* license info. You should avoid requiring a contributor license agreement, though it is possible.

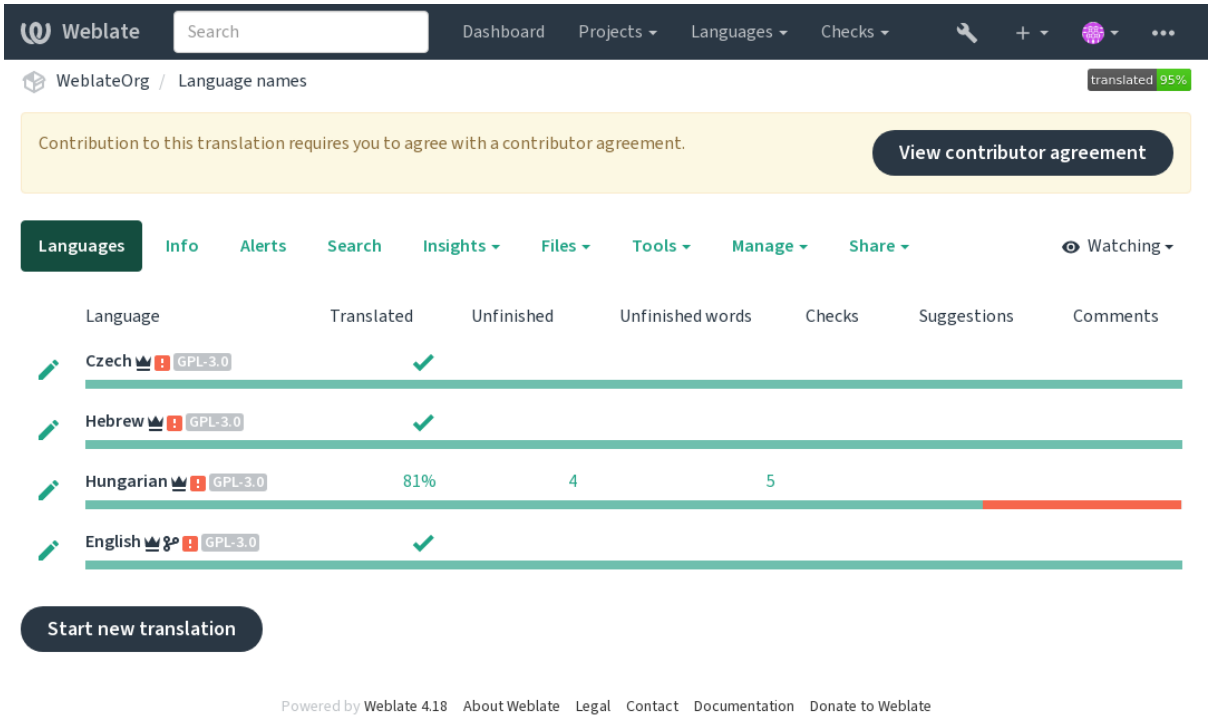
2.10.1 License info

Upon specifying license info (license name and URL), this info is shown in the translation info section of the respective *Configuración de componentes*.

Usually this is best place to post licensing info if no explicit consent is required. If your project or translation is not libre you most probably need prior consent.

2.10.2 Acuerdo de contribuidor

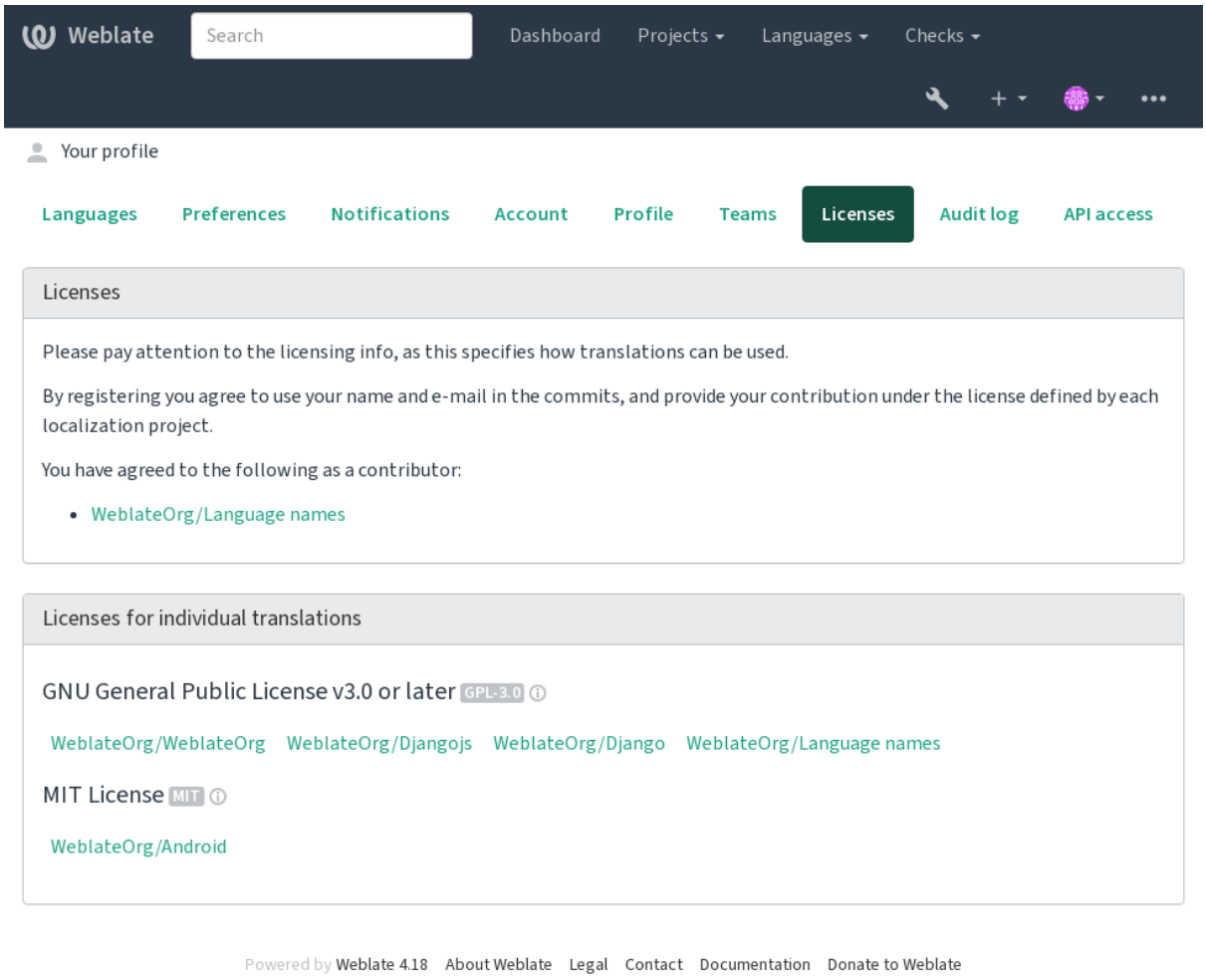
If you specify a contributor license agreement, only users who have agreed to it will be able to contribute. This is a clearly visible step when accessing the translation:



The entered text is formatted into paragraphs and external links can be included. HTML markup can not be used.

2.10.3 User licenses

Any user can review all translation licenses of all public projects on the instance from their profile:



2.11 Proceso de traducción

2.11.1 Votar sugerencias

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 *Configuración de componentes* 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).

Nota: 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*.

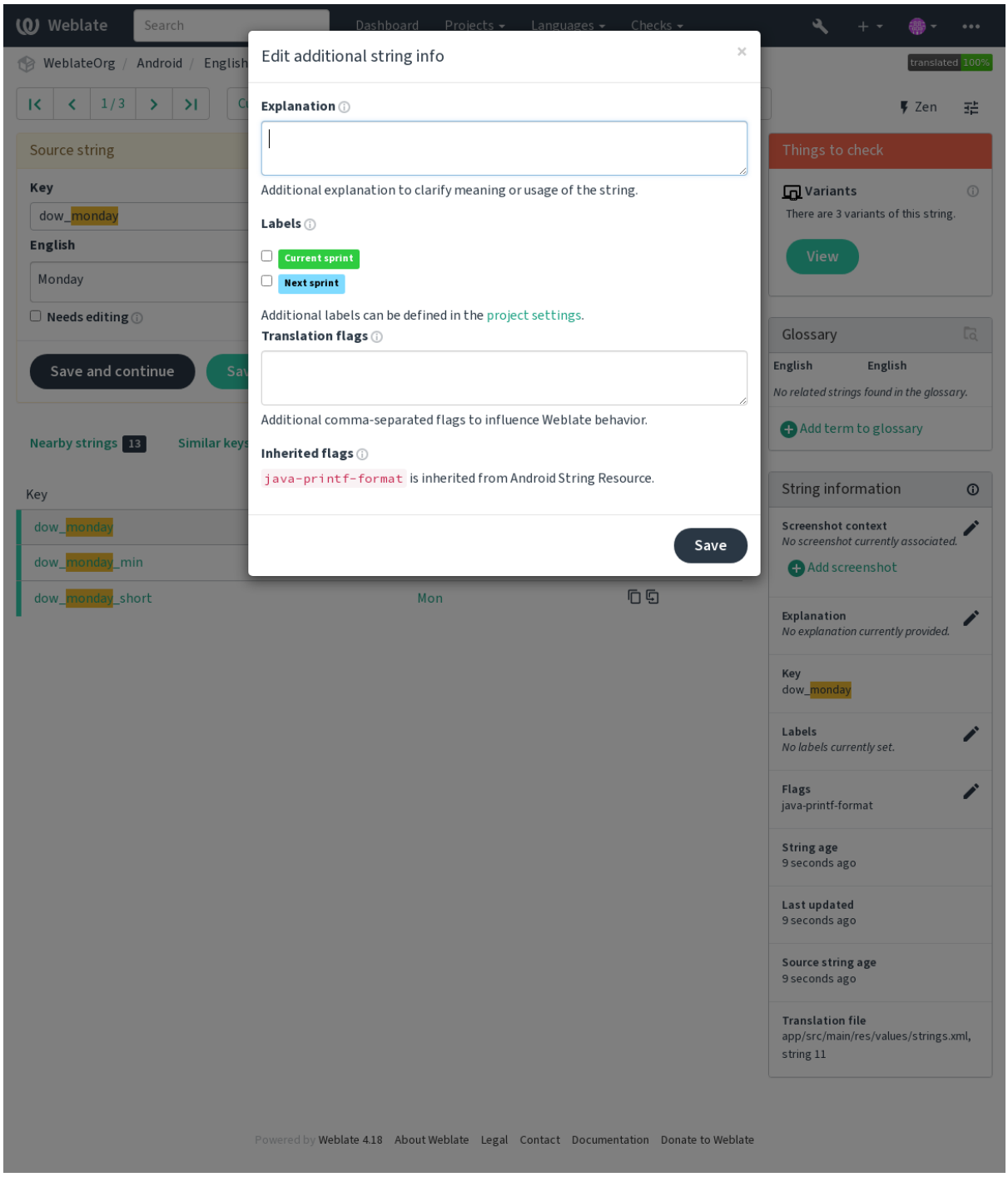
You can combine these with *access control* into one of the following setups:

- 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 Información adicional sobre las cadenas de origen

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:



Access this directly from the translation interface by clicking the «Edit» icon next to *Screenshot context* or *Flags*.

Weblate
Search
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Czech / Translate
translated 96%
11 / 26
All strings
Position and priority
Zen
Translation
Explanation
Help text for automatic translation tool
English
Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.
Czech
Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.
Needs editing
169/1570 · 157
Save and continue
Save and stay
Suggest
Skip
Nearby strings 26
Comments
Automatic suggestions
Other languages 4
History
Context
English
Czech
Actions
Files
Soubory
Automatic translation
Automatický překlad
Add new translation string
Add new translation string
Translation status
Stav překladu
Singular
%(count)s word
Plural
%(count)s words
One
%(count)s slovo
Few
%(count)s slova
Many
%(count)s slov
Other components
Další součásti
Translation file
Soubor s překladem
Download
Stáhnout
Browse all translation changes
Procházet všechny změny v překladu.
Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.
Automatický překlad použije stávající překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.
Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.
Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.
You can add new translation string here, it will automatically appear in all translations.
Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.
The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.
Nahraný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.
The uploaded file will be merged with the current translation.
Nahraný soubor bude sloučen se stávajícími překlady.
The fulltext search might not work properly as the fulltext index for this translation is not yet up to date.
Fulltextové vyhledávání nemusí fungovat správě, protože fulltextový index pro tento překlad ještě není plně zpracován.
Review
Kontrola
Review translations touched by other users.
Zkontrolovat překlady od ostatních uživatelů.
Start review
Začít kontrolu
Percent
Procenta
Total
Celkem
Failing check
Neúspěšných kontrol
Last activity
Poslední aktivita
Last change
Poslední změna
Last author
Poslední autor
Question for a mathematics-based CAPTCHA, the %s is an arithmetic problem
What is %s?
Kolik to je?
The string uses three dots (...) instead of an ellipsis character (…)
Glossary
English
Czech
machine translation
strojový překlad
WeblateOrg
project
projekt
WeblateOrg
Add term to glossary
String information
Screenshot context
No screenshot currently associated.
Add screenshot
Explanation
Help text for automatic translation tool
Labels
No labels currently set.
Flags
No flags currently set.
Source string location
weblate/templates/translation.html:212
String age
3 seconds ago
Last updated
3 seconds ago
Source string age
3 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 11
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Ver también:

Ubicación de cadena de origen, *Descripción de cadena de origen*, *Contexto*

Priorización de cadenas

String priority can be changed to offer higher priority strings for translation earlier by using the `priority` flag.

Consejo: This can be used to order the flow of translation in a logical manner.

Ver también:

Comprobaciones de calidad

Indicadores de traducción

Distinto en la versión 3.3: Previously called *Quality checks flags*, it no longer configures only checks.

Customization of quality checks and other Weblate behavior, see *Personalizar el comportamiento mediante indicadores*.

The string flags are also inherited from the *Indicadores de traducción* at *Configuración de componentes* and flags from the translation file (see *Formatos de archivo admitidos*).

Ver también:

Comprobaciones de calidad, *Personalizar el comportamiento mediante indicadores*

Explicación

Distinto en la versión 4.1: En versiones anteriores se le ha llamado *Extra context*.

Distinto en la versión 4.18: Support for synching explanation with a file was introduced.

Utilice la explicación para aclarar el ámbito o la utilización de la traducción. Puede utilizar Markdown para incluir enlaces y dar formato.

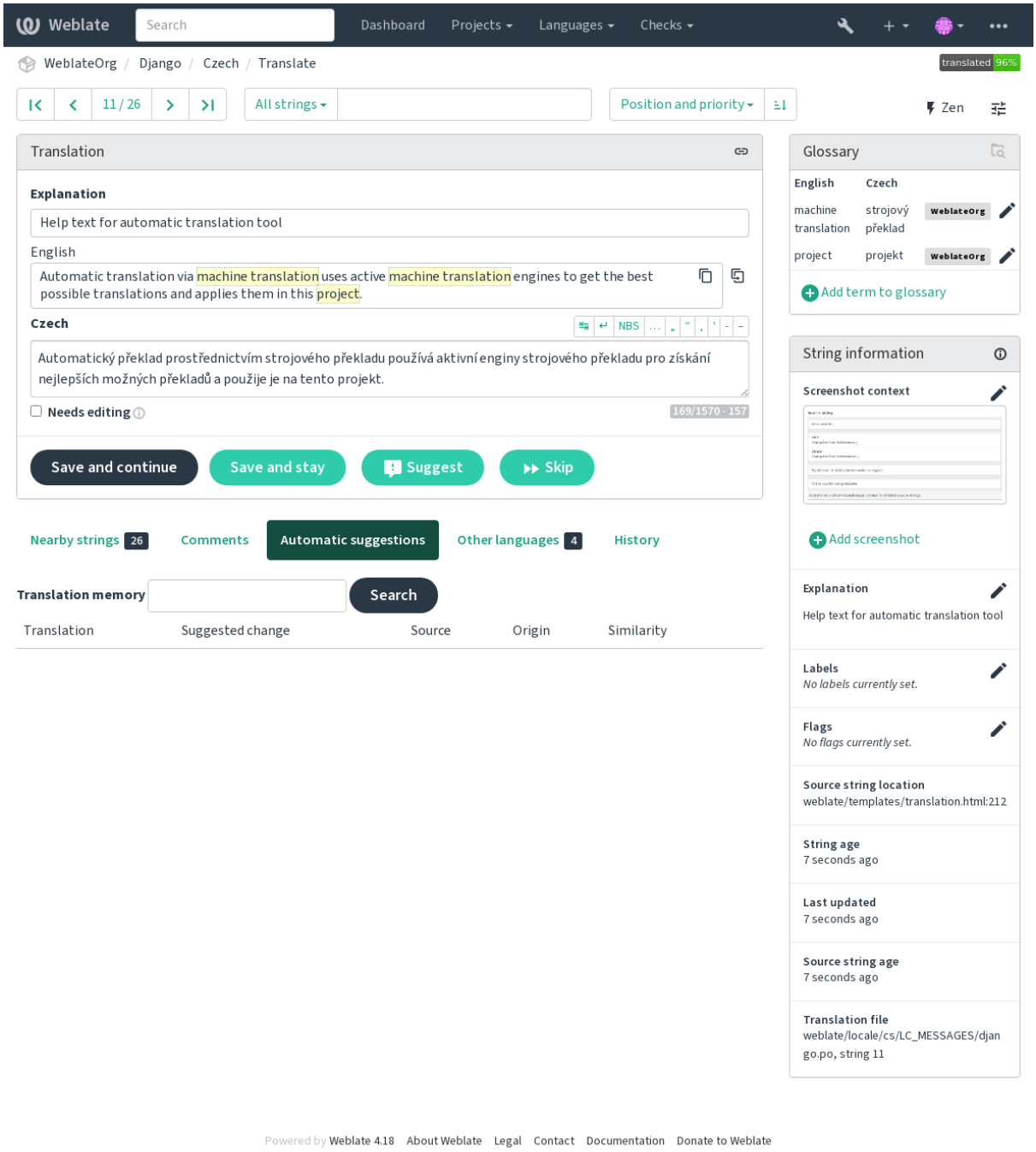
Some file formats support storing explanation within the file, see *Explicación*.

Consejo: Weblate can also display description present in the translation file for some formats, see *Descripción de cadena de origen*.

Contexto visual para cadenas

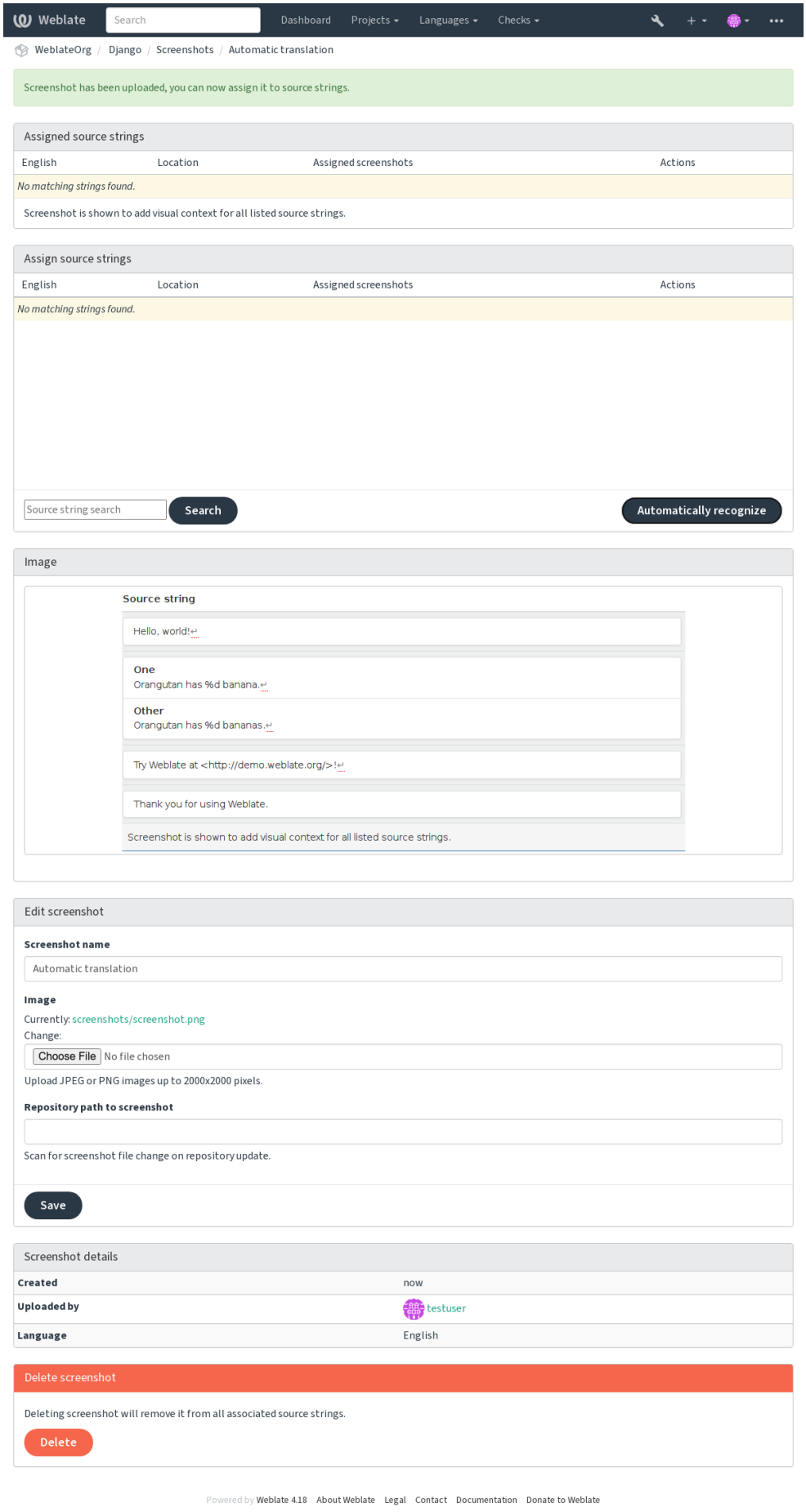
You can upload a screenshot showing a given source string in use within your program. This helps translators understand where it is used, and how it should be translated.

La captura cargada se muestra en la barra lateral de contexto de la traducción:



In addition to *Información adicional sobre las cadenas de origen*, screenshots have a separate management interface under the *Tools* menu. Upload screenshots, assign them to source strings manually, or use optical character recognition (OCR) to do so by pressing the *Automatically recognize* button.

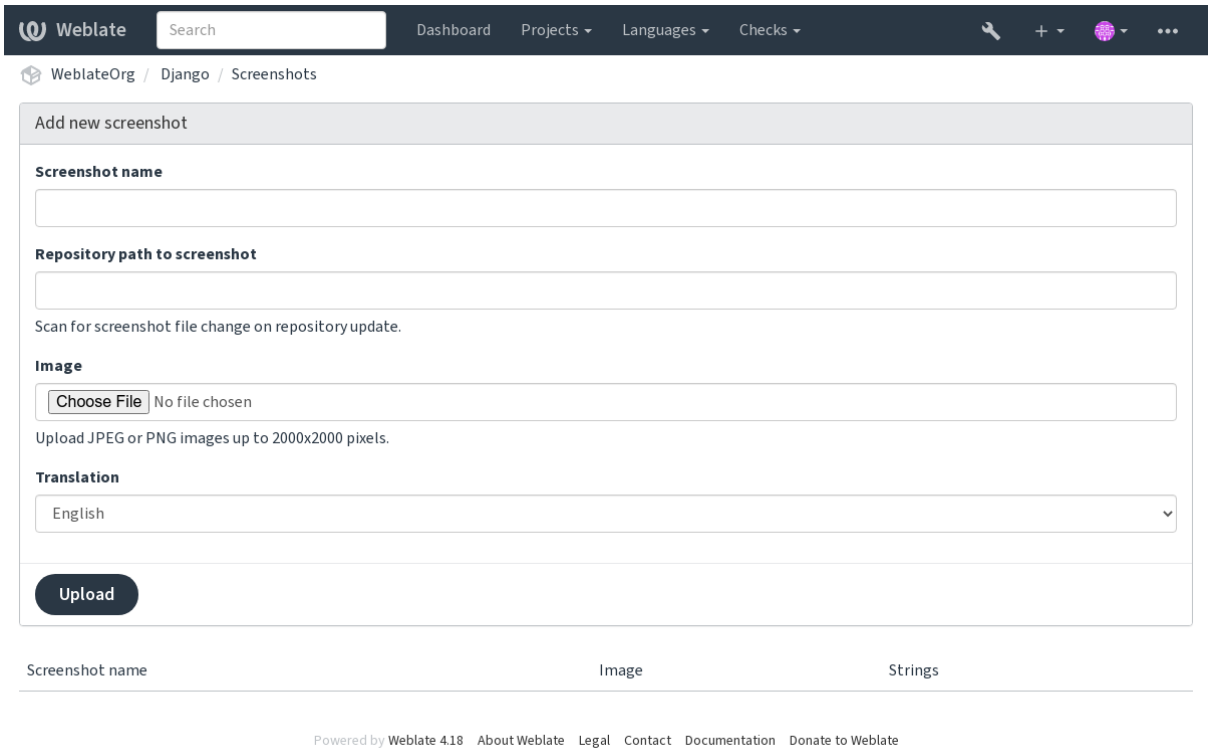
Once a screenshot is uploaded, this interface handles management and source string association:


Weblate
Search
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.
Repository path to screenshot
Scan for screenshot file change on repository update.
Save
Screenshot details
Created
now
Uploaded by
testuser
Language
English
Delete screenshot
Deleting screenshot will remove it from all associated source strings.
Delete
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

You can add or update screenshots directly from your Version Control System (VCS) repository.

To enable this feature, you can either set a screenshot file mask when creating a component, which will be monitored for updates in the repository, or you can add or update screenshots when uploading them manually.

When the repository is updated, the system will automatically scan for changes. Existing screenshots in the repository will be updated, and new screenshots matching the specified screenshot file mask will be added to the component.



Ver también:

Screenshot file mask

2.12 Comprobaciones y correcciones

2.12.1 Correcciones automáticas personalizadas

También puede implementar sus propias correcciones automáticas además de las estándar e incluirlas en *AUTOFIX_LIST*.

Las correcciones automáticas son potentes pero pueden causar daños; tenga cuidado al escribir una.

Por ejemplo, la siguiente corrección automática reemplazaría cada aparición de la cadena `foo` en una traducción por `bar`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy

from weblate.trans.autofixes.base import AutoFix


class ReplaceFooWithBar(AutoFix):
```

(continué en la próxima página)

(proviene de la página anterior)

```
    """Replace foo with bar."""

    name = gettext_lazy("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False
```

Para instalar comprobaciones personalizadas, proporcione una ruta completa a la clase de Python en el parámetro *AUTOFIX_LIST*, véase *Controles de calidad personalizados, complementos y autocorrecciones*.

2.12.2 Personalizar el comportamiento mediante indicadores

You can fine-tune Weblate's behavior by using flags. The flags provide visual feedback to the translators and help them to improve their translation. This can be done on the source string level (see *Información adicional sobre las cadenas de origen*), or in the *Configuración de componentes* (*Indicadores de traducción*). Some file formats also allow to specify flags directly in the format (see *Formatos de archivo admitidos*).

The flags are comma-separated; if they have parameters, they are separated with colon. You can use quotes to include whitespaces or special characters in the string. For example:

```
placeholders:"special:value":"other value", regex:.*
```

Se aceptan tanto las comillas simples como las dobles, los caracteres especiales se escapan utilizando la barra invertida:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

```
placeholders:r"^#*"
```

To verify that translators do not change the heading of a Markdown document: A failing check will be triggered if the string "### Index" is translated as "# Indice"

```
placeholders:r"\]\([^h].*?\)"
```

To make sure that internal links are not being translated (i.e. *[test](../checks)* does not become *[test](../chequeos)*.

Aquí hay una lista de banderas actualmente aceptadas:

rst-text
: Tratar un texto como un documento reStructuredText, afecta a *Traducción no modificada*.

dos-eol
: Utiliza los marcadores de fin de línea de DOS en lugar de los de Unix (`r\n` instead of `\n`).

read-only
: La cadena es de solo lectura y no debe editarse en Weblate; vea *Cadenas de solo lectura*.

terminology
: Used in *Glosario*. Copies the string into all glossary languages so it can be used consistently in all translations. Also useful in combination with `read-only`, for example in product names.

priority:N
: Prioridad de la cadena. Las cadenas de mayor prioridad se presentan primero para su traducción. La prioridad por defecto es 100, cuanto mayor sea la prioridad de una cadena, antes se ofrecerá para su traducción.

max-length:N
: Limita la longitud máxima de una cadena a N caracteres, véase *Longitud máxima de la traducción*.

xml-text
: Tratar el texto como un documento XML, afecta a *Sintaxis XML* y *Marcación XML*.

font-family:NOMBRE
Definir la familia de fuentes para las comprobaciones de renderización, ver *Gestionar tipos de letra*.

font-weight:PESO
Definir el peso de la fuente para las comprobaciones de renderización, ver *Gestionar tipos de letra*.

font-size:TAMAÑO
Definir el tamaño de la fuente para las comprobaciones de renderización, ver *Gestionar tipos de letra*.

font-spacing:ESPACIADO
Definir el espaciado de las letras para las comprobaciones de renderización, ver *Gestionar tipos de letra*.

icu-flags:FLAGS
Defina las banderas para personalizar el comportamiento del control de calidad *MessageFormat de ICU*.

icu-tag-prefix:PREFIX
Establezca un prefijo requerido para las etiquetas XML para el control de calidad *MessageFormat de ICU*.

placeholders:NOMBRE:NOMBRE2:...
Cadenas sustitutivas esperadas en la traducción; vea *Sustitutivos*.

replacements:FROM:TO:FROM2:TO2...
Reemplazos a realizar cuando se comprueban los parámetros de texto resultantes (por ejemplo en *Tamaño máximo de la traducción* o *Longitud máxima de la traducción*). El caso de uso típico para esto es expandir los placeables para asegurar que el texto encaje incluso con valores largos, por ejemplo: `replacements:%s:"John Doe"`.

variantes:FUENTE
Marcar esta cadena como una variante de cadena con origen coincidente. Ver variants.

regex:REGEX
Expresión regular utilizada para relacionar traducciones; vea *Expresión regular*.

forbidden
Indica una traducción prohibida en un glosario, ver *Traducciones prohibidas*.

strict-same
Hace que «Traducción no modificada» evite la lista de palabras incorporada; vea *Traducción no modificada*.

check-glossary
Activar el control de calidad *No se ajusta al glosario*.

angularjs-format
Activar el control de calidad *Cadena de interpolación de AngularJS*.

c-format
Activar el control de calidad *Formato C*.

c-sharp-format
Activar el control de calidad *Formato C#*.

es-format
Activar el control de calidad *Literales de plantilla ECMAScript*.

i18next-interpolation
Activar el control de calidad *Interpolación con i18next*.

icu-message-format
Habilite el control de calidad *MessageFormat de ICU*.

java-printf-format
Habilite el control de calidad *Formato Java*.

java-format
Activar el control de calidad *MessageFormat de Java*.

javascript-format
Activar el control de calidad *Formato JavaScript*.

lua-format
Activar el control de calidad *Formato Lua*.

object-pascal-format
Activar el control de calidad *Formato de Object Pascal*.

percent-placeholders
Activar el control de calidad *Sustitutivos con signo de porcentaje*.

perl-format
Activar el control de calidad *Formato Perl*.

php-format
Activar el control de calidad *Formato PHP*.

python-brace-format
Activar el control de calidad *Formato de llaves de Python*.

python-format
Activar el control de calidad *Formato Python*.

qt-format
Activa el control de calidad *Formato Qt*.

qt-plural-format
Activa el control de calidad *Formato de plurales de Qt*.

ruby-format
Activa el control de calidad *Formato Ruby*.

scheme-format
Activa el control de calidad *Formato Scheme*.

vue-format
Activa el control de calidad *Formato de Vue I18n*.

md-text
Treat text as a Markdown document, and provide Markdown syntax highlighting on the translation text area. Enables *Enlaces de Markdown*, *Referencias de Markdown*, and *Sintaxis de Markdown* quality checks.

case-insensitive
Ajustar el comportamiento de las comprobaciones para que no distingan entre mayúsculas y minúsculas. Actualmente sólo afecta a la comprobación de calidad *Sustitutivos*.

safe-html
Activa el control de calidad *HTML inseguro*.

url
La cadena debe consistir de solo una URL. Activa el control de calidad *URL*.

ignore-all-checks
Ignora todos los controles de calidad.

ignore-bbcode
Saltar el control de calidad *Marcación BBCode*.

ignore-duplicate
Omite el control de calidad *Palabras consecutivas duplicadas*.

ignore-check-glossary
Omite el control de calidad *No se ajusta al glosario*.

ignore-double-space
Omite el control de calidad *Espacio duplicado*.

ignore-angularjs-format
Omite el control de calidad *Cadena de interpolación de AngularJS*.

ignore-c-format
Omite el control de calidad *Formato C*.

ignore-c-sharp-format
Omite el control de calidad *Formato C#*.

ignore-es-format
Omite el control de calidad *Literales de plantilla ECMAScript*.

ignore-i18next-interpolation
Omite el control de calidad *Interpolación con i18next*.

ignore-icu-message-format
Omita el control de calidad *MessageFormat de ICU*.

ignore-java-printf-format
Omita el control de calidad *Formato Java*.

ignore-java-format
Omite el control de calidad *MessageFormat de Java*.

ignore-javascript-format
Omite el control de calidad *Formato JavaScript*.

ignore-lua-format
Omita el control de calidad *Formato Lua*.

ignore-object-pascal-format
Omita el control de calidad *Formato de Object Pascal*.

ignore-percent-placeholders
Pasa por alto la comprobación de calidad *Sustitutivos con signo de porcentaje*.

ignore-perl-format
Omita el control de calidad *Formato Perl*.

ignore-php-format
Omita el control de calidad *Formato PHP*.

ignore-python-brace-format
Omita el control de calidad *Formato de llaves de Python*.

ignore-python-format
Omita el control de calidad *Formato Python*.

ignore-qt-format
Omita el control de calidad *Formato Qt*.

ignore-qt-plural-format
Omita el control de calidad *Formato de plurales de Qt*.

ignore-ruby-format
Omita el control de calidad *Formato Ruby*.

ignore-scheme-format
Omita el control de calidad *Formato Scheme*.

ignore-vue-format
Omita el control de calidad *Formato de Vue I18n*.

ignore-translated
Omita el control de calidad *Se había traducido*.

ignore-inconsistent
Omita el control de calidad *Incoherente*.

ignore-kashida
Omite la comprobación de calidad *Kashida utilizado*.

ignore-md-link
Omite la comprobación de calidad *Enlaces de Markdown*.

ignore-md-reflink
Omite la comprobación de calidad *Referencias de Markdown*.

ignore-md-syntax
Omite la comprobación de calidad *Sintaxis de Markdown*.

ignore-max-length
Omite la comprobación de calidad *Longitud máxima de la traducción*.

ignore-max-size
Omite la comprobación de calidad *Tamaño máximo de la traducción*.

ignore-escaped-newline
Omite la comprobación de calidad *\n desiguales*.

ignore-end-colon
Omite la comprobación de calidad *Dos puntos desiguales*.

ignore-end-ellipsis
Omite la comprobación de calidad *Puntos suspensivos desiguales*.

ignore-end-exclamation
Omite la comprobación de calidad *Signo de exclamación desigual*.

ignore-end-stop
Omite el control de calidad *Punto final desigual*.

ignore-end-question
Omite el control de calidad *Signo de interrogación desigual*.

ignore-end-semicolon
Omite la comprobación de calidad *Punto y coma desigual*.

ignore-newline-count
Omite la comprobación de calidad *Saltos de renglón desiguales*.

ignore-plurals
Omite el control de calidad *Faltan plurales*.

ignore-placeholders
Pasa por alto la comprobación de calidad *Sustitutivos*.

ignore-punctuation-spacing
Omite la comprobación de calidad *Espaciado de puntuación*.

ignore-regex
Omite la comprobación de calidad *Expresión regular*.

ignore-reused
Skip the *Reused translation* quality check.

ignore-same-plurals
Omite la comprobación de calidad *Mismos plurales*.

ignore-begin-newline
Omite la comprobación de calidad *Salto de renglón al inicio*.

ignore-begin-space
Omite la comprobación de calidad *Espacios iniciales*.

ignore-end-newline
Omite la comprobación de calidad *Salto de renglón al final*.

ignore-end-space
Omite la comprobación de calidad *Espacio al final*.

ignore-same
Omite la comprobación de calidad *Traducción no modificada*.

ignore-safe-html
Omite la comprobación de calidad *HTML inseguro*.

ignore-url
Omite la comprobación de calidad *URL*.

ignore-xml-tags
Omite la comprobación de calidad *Marcación XML*.

ignore-xml-invalid
Omite la comprobación de calidad *Sintaxis XML*.

ignore-zero-width-space
Omite la comprobación de calidad *Espacio de anchura cero*.

ignore-ellipsis
Omite la comprobación de calidad *Puntos suspensivos*.

ignore-icu-message-format-syntax
Omite la comprobación de calidad *Sintaxis MessageFormat de ICU*.

ignore-long-untranslated
Omite la comprobación de calidad *Largamente no traducida*.

ignore-multiple-failures
Omite la comprobación de calidad *Varias comprobaciones fallidas*.

ignore-unnamed-format
Omite la comprobación de calidad:ref:*check-unnamed-format*.

ignore-optional-plural
Omite la comprobación de calidad *No pluralizada*.

Nota: Generalmente la regla se denomina `ignore-*` para cualquier comprobación, usa su identificador, así puedes utilizarlo incluso para tus comprobaciones personalizadas.

Estas flags se entienden tanto en la configuración de *Configuración de componentes*, por la configuración de la cadena de origen y en el propio archivo de traducción (por ejemplo en GNU gettext).

2.12.3 Forzar comprobaciones

Nuevo en la versión 3.11.

Puede configurar una lista de comprobaciones que no pueden ser ignoradas estableciendo *Comprobaciones obligatorias* en *Configuración de componentes*. Cada una de las comprobaciones de la lista no se puede descartar en la interfaz de usuario y cualquier cadena que falle esta comprobación se marca como *Needs editing* (ver *Estados de traducción*).

Nota: La activación de la comprobación no la habilita automáticamente. La comprobación puede activarse añadiendo la bandera correspondiente a las banderas de cadenas o componentes.

Ver también:

Información adicional sobre las cadenas de origen, *Indicadores de traducción*

2.12.4 Gestionar tipos de letra

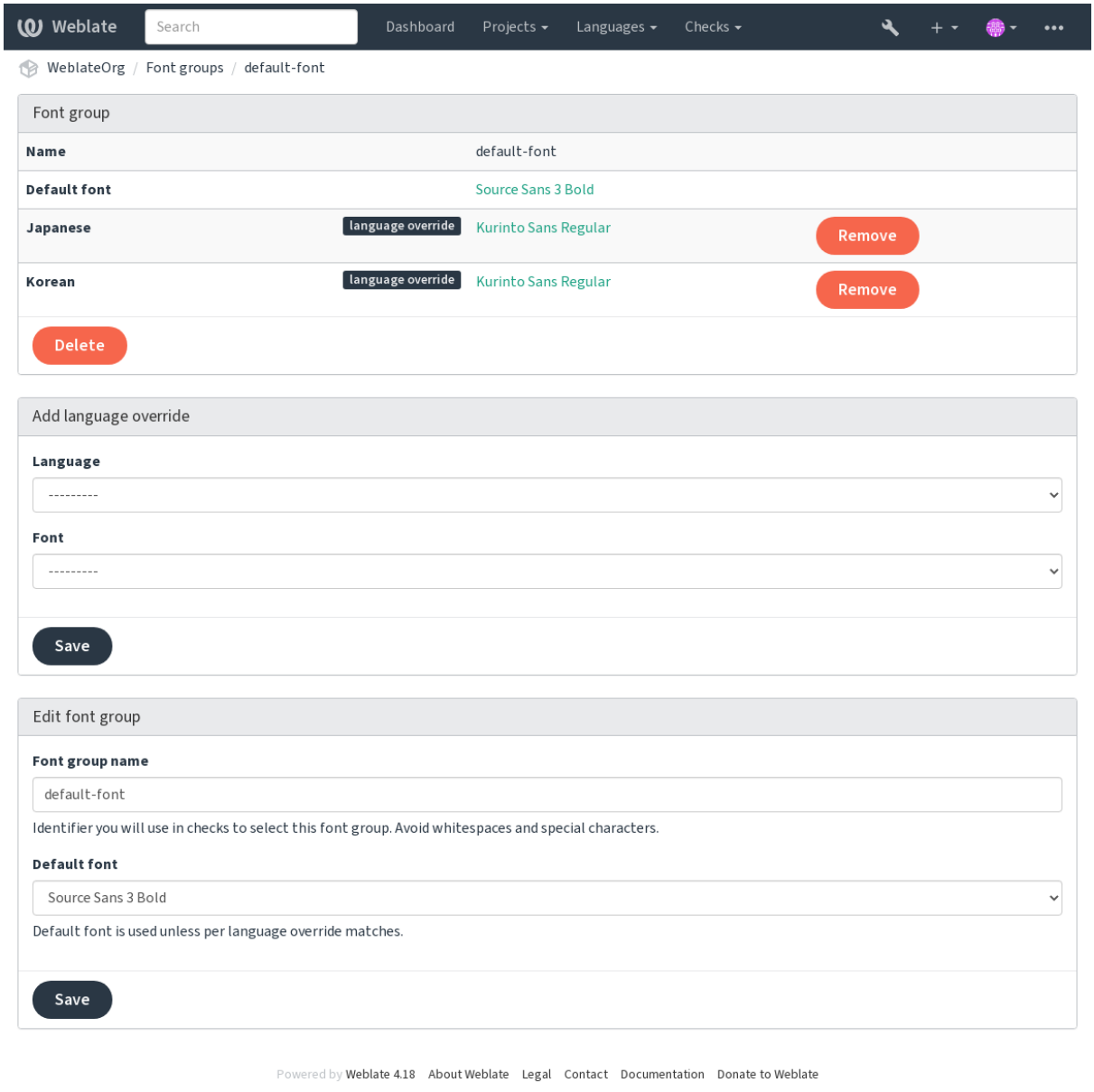
Nuevo en la versión 3.7.

Consejo: Las fuentes cargadas en Weblate se usan únicamente a efectos de comprobación *Tamaño máximo de la traducción*, no tienen efecto en la interfaz de usuario de Weblate.

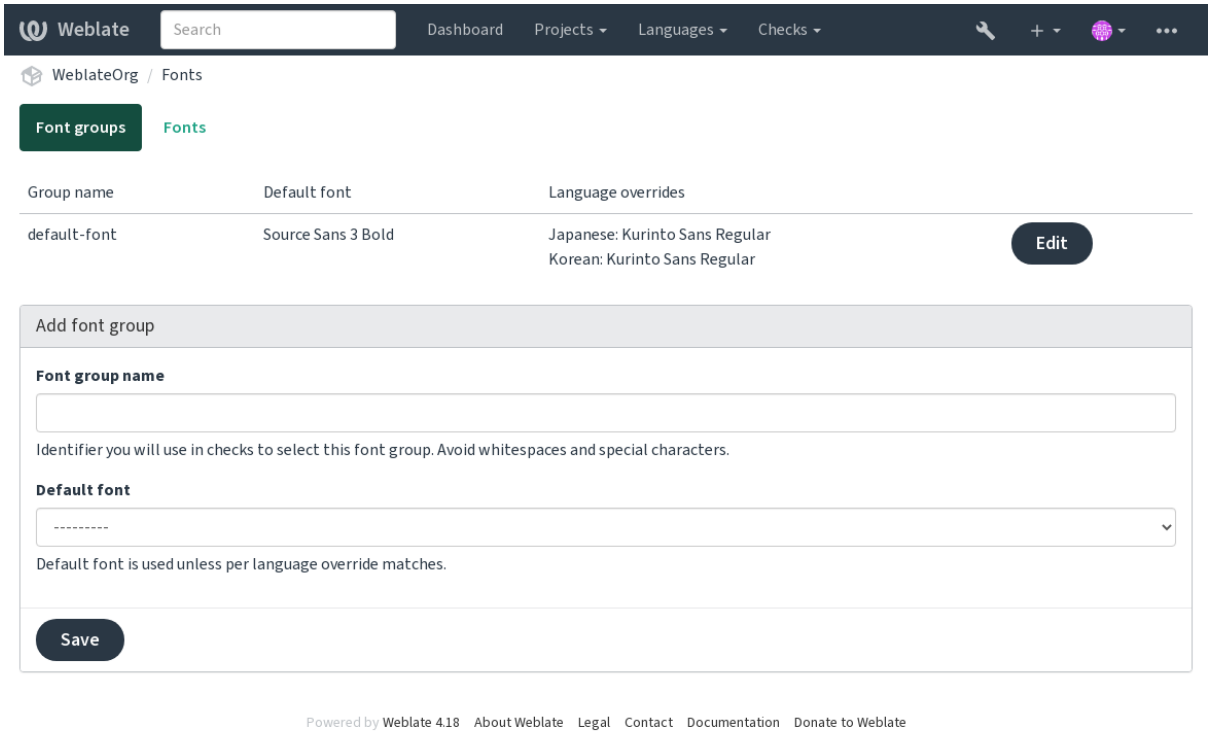
La comprobación *Tamaño máximo de la traducción* utilizada para calcular las dimensiones del texto renderizado necesita que la fuente se cargue en Weblate y se seleccione mediante una flag de traducción (ver *Personalizar el comportamiento mediante indicadores*).

La herramienta de gestión de fuentes de Weblate en *Fonts* bajo el menú *Manage* de tu proyecto de traducción proporciona una interfaz para cargar y gestionar las fuentes. Se pueden cargar fuentes TrueType u OpenType, configurar grupos de fuentes y utilizarlas en la comprobación.

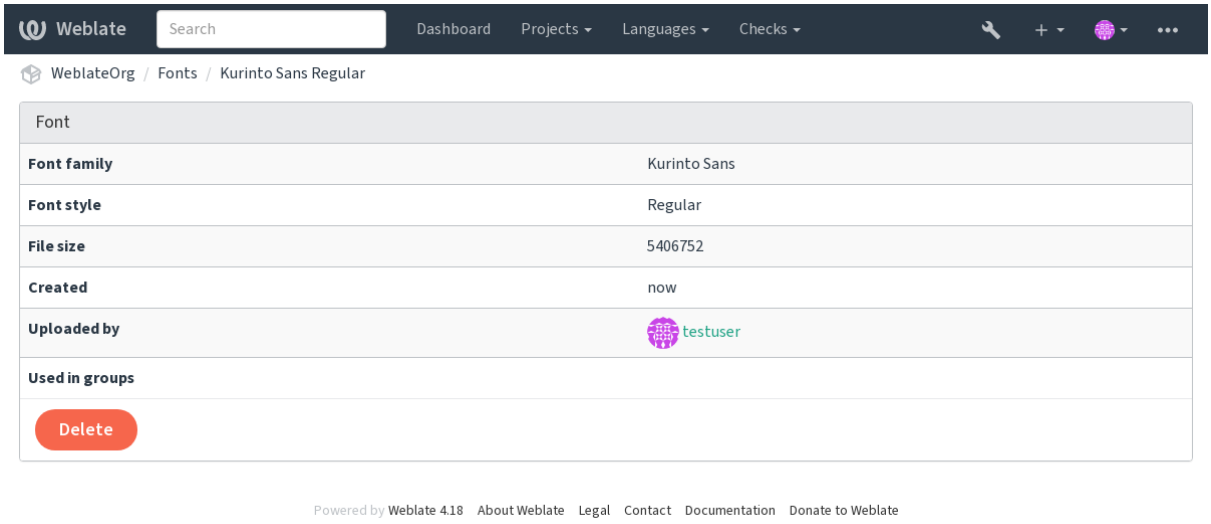
Los grupos de fuentes permiten definir diferentes fuentes para distintos idiomas, lo que suele ser necesario para los idiomas no latinos:



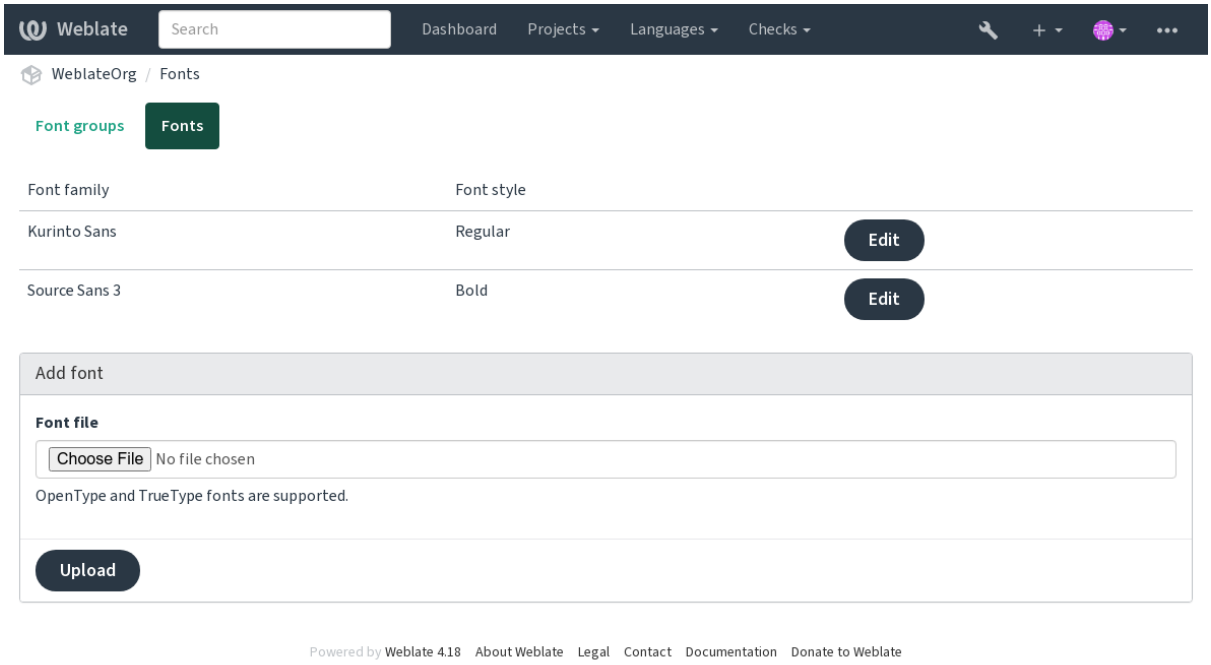
Los grupos de fuentes se identifican por su nombre, que no puede contener espacios en blanco ni caracteres especiales, para que pueda utilizarse fácilmente en la definición de la comprobación:



El tipo de letra y el estilo se reconocen automáticamente después de cargarlos:



Puedes tener varias fuentes cargadas en Weblate:



Para utilizar las fuentes que comprueban la longitud de la cadena, pásale los indicadores adecuados (ver *Personalizar el comportamiento mediante indicadores*). Posiblemente necesitarás los siguientes:

max-size:500
: Define la anchura máxima en píxeles.

font-family:ubuntu
: Define el grupo de fuentes a utilizar especificando su identificador.

font-size:22
: Define el tamaño de la fuente en pixeles.

2.12.5 Escribir los propios controles

Se han incorporado una amplia gama de controles de calidad, (ver *Comprobaciones de calidad*), aunque es posible que no cubran todo lo que quieres comprobar. Se puede ajustar la lista de comprobaciones realizadas mediante `CHECK_LIST`, y también puedes añadir controles personalizados.

1. Subclase del *weblate.checks.Check*
2. Defina algunos atributos.
3. Implementa el ``check``(si quieres tratar los plurales en tu código) o el método ``check_single``(que lo hace por ti).

He aquí algunos ejemplos:

Para instalar controles personalizados, proporciona una ruta muy clara de clase Python en `CHECK_LIST`, ver *Controles de calidad personalizados, complementos y autocorrecciones*.

Comprueba que el texto de la traducción no contiene «foo»

Esta es una comprobación bastante simple que se limita a revisar si a la traducción le falta la cadena «tal».

```
# 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

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 = gettext_lazy("Foo check")

    # Description for failing check
    description = gettext_lazy("Your translation is foo")

    # Real check code
    def check_single(self, source, target, unit):
        return "foo" in target
```

Comprueba que los plurales del texto traducido al checo son diferentes

Comprobación que utiliza la información de idioma para cerciorarse de que las dos formas de plural en checo no sean iguales.

```
# 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

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 = gettext_lazy("Foo check")

    # Description for failing check
    description = gettext_lazy("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]
```

(continué en la próxima página)

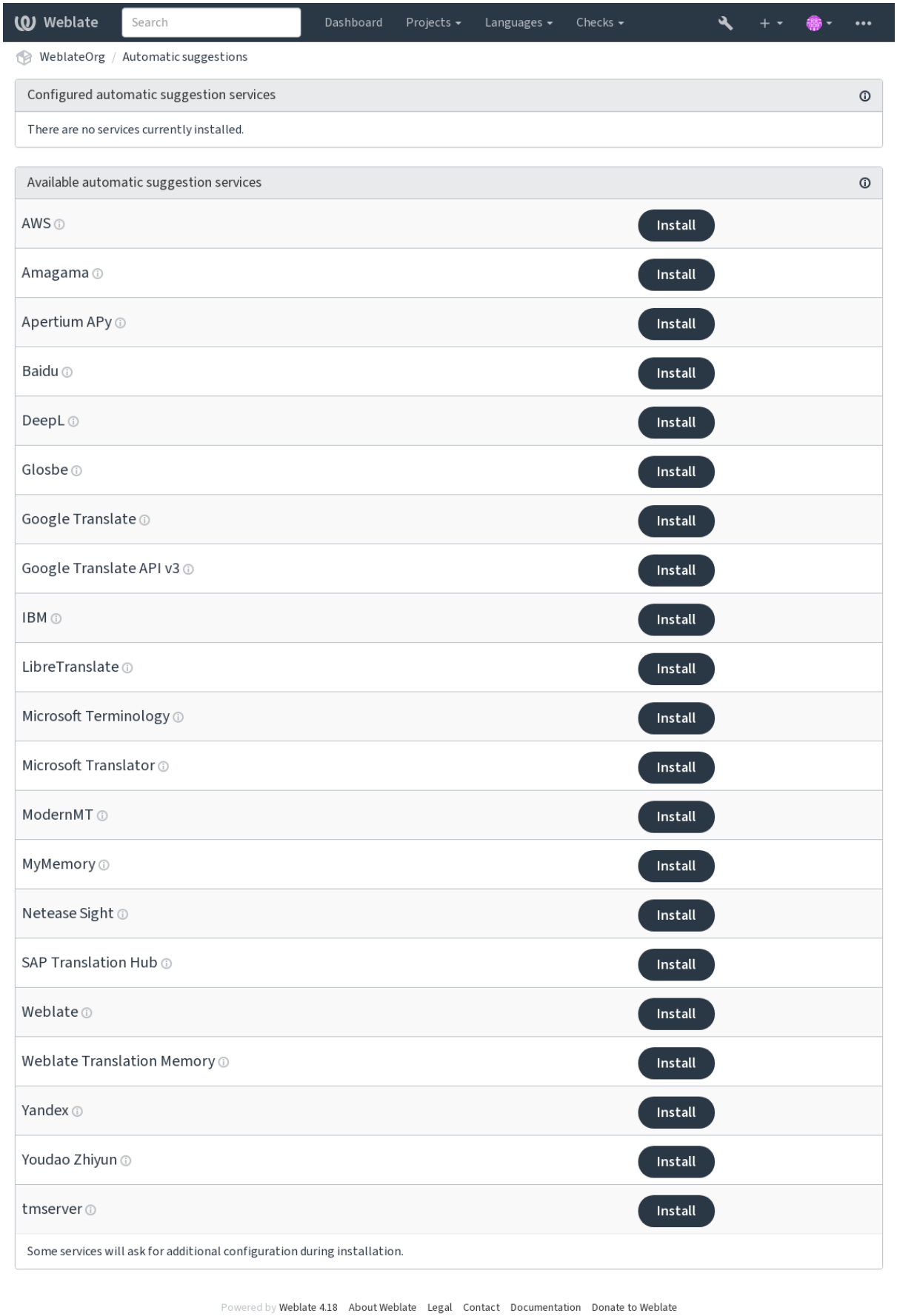
(proviene de la página anterior)

```
        return False

    def check_single(self, source, target, unit):
        """We don't check target strings here."""
        return False
```

2.13 Configuración de las sugerencias automáticas

Distinto en la versión 4.13: Prior to Weblate 4.13, the services were configured in the *Configuración*.

The support for several machine translation and translation memory services is built-in. Each service can be turned on by the administrator for whole site or at the project settings:


Weblate
Search
Dashboard
Projects
Languages
Checks
WeblateOrg / Automatic suggestions
Configured automatic suggestion services
There are no services currently installed.
Available automatic suggestion services
AWS
Amagama
Apertium APy
Baidu
DeepL
Glosbe
Google Translate
Google Translate API v3
IBM
LibreTranslate
Microsoft Terminology
Microsoft Translator
ModernMT
MyMemory
Netease Sight
SAP Translation Hub
Weblate
Weblate Translation Memory
Yandex
Youdao Zhiyun
tmserver
Install
Some services will ask for additional configuration during installation.
Powered by Weblate 4.18
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Nota: They come subject to their terms of use, so ensure you are allowed to use them how you want.

Los servicios se traducen del idioma de origen según lo configurado en *Configuración de componentes*, consulte *Idioma del código fuente*.

Ver también:

Sugerencias automáticas

2.13.1 Amagama

Identificación del servicio
`amagama`

Configuración
This service has no configuration.

Special installation of *tmserver* run by the authors of Virtaal.

Ver también:

Installing amaGama, Amagama, amaGama Translation Memory

2.13.2 APy de Apertium

Identificación del servicio
`apertium-apy`

Configuración

`url`	URL DE LA APLICACIÓN

A libre software machine translation platform providing translations to a limited set of languages.

The recommended way to use Apertium is to run your own Apertium-APy server.

Ver también:

Apertium website, Apertium APy documentation

2.13.3 AWS

Nuevo en la versión 3.1.

Identificación del servicio
`aws`

Configuración

`key`	Identificador de la clave de acceso
`secret`	Clave secreta de la aplicación
`region`	Nombre de la región

Amazon Translate is a neural machine translation service for translating text to and from English across a breadth of supported languages.

Ver también:

Amazon Translate Documentation

2.13.4 Baidu

Nuevo en la versión 3.2.

Identificación del servicio
`baidu`

Configuración

`key`	Identificación del cliente
`secret`	Secreto del cliente

Machine translation service provided by Baidu.

This service uses an API and you need to obtain an ID and API key from Baidu to use it.

Ver también:

Baidu Translate API

2.13.5 DeepL

Identificación del servicio
`deepl`

Configuración

`url`	URL DE LA APLICACIÓN
`key`	Clave de la aplicación

DeepL is paid service providing good machine translation for a few languages. You need to purchase *DeepL API* subscription or you can use legacy *DeepL Pro (classic)* plan.

API URL to use with the DeepL service. At the time of writing, there is the v1 API as well as a free and a paid version of the v2 API.

`https://api.deepl.com/v2/` (default in Weblate)
Is meant for API usage on the paid plan, and the subscription is usage-based.

`https://api-free.deepl.com/v2/`
Is meant for API usage on the free plan, and the subscription is usage-based.

`https://api.deepl.com/v1/`
Is meant for CAT tools and is usable with a per-user subscription.

Previously Weblate was classified as a CAT tool by DeepL, so it was supposed to use the v1 API, but now is supposed to use the v2 API. Therefore it defaults to v2, and you can change it to v1 in case you have an existing CAT subscription and want Weblate to use that.

The easiest way to find out which one to use is to open an URL like the following in your browser:

https://api.deepl.com/v2/translate?text=Hello&target_lang=FR&auth_key=XXX

Replace the XXX with your auth_key. If you receive a JSON object which contains «Bonjour», you have the correct URL; if not, try the other three.

Weblate supports DeepL formality, it will choose matching one based on the language (for example, there is `de@formal` and `de@informal`).

Ver también:

DeepL website, DeepL pricing, DeepL API documentation

2.13.6 Glosbe

Identificación del servicio
`glosbe`

Configuración
This service has no configuration.

Free dictionary and translation memory for almost every living language.

The API is gratis to use, but usage of the translations is subject to the license of the used data source. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Ver también:

Glosbe website

2.13.7 Google Translate

Identificación del servicio
`google-translate`

Configuración

`key`	Clave de la aplicación

Machine translation service provided by Google.

This service uses the Google Translation API, and you need to obtain an API key and turn on billing in the Google API console.

Ver también:

Google translate documentation

2.13.8 API de Google Translate v3

Identificación del servicio
`google-translate-api-v3`

Configuración

`credentials`	Información de la cuenta del servicio Google Translate
`project`	Proyecto de Google Translate
`location`	Ubicación de Google Translate

Machine translation service provided by Google Cloud services.

Ver también:

Google translate documentation, Authenticate to Cloud services using client libraries, Creating Google Translate project, Google Cloud App Engine locations

2.13.9 LibreTranslate

Nuevo en la versión 4.7.1.

Identificación del servicio
`libretranslate`

Configuración

`url`	URL DE LA APLICACIÓN
`key`	Clave de la aplicación

LibreTranslate is a free and open-source service for machine translations. The public instance requires an API key, but LibreTranslate can be self-hosted and there are several mirrors available to use the API for free.

`https://libretranslate.com/` (official public instance)
Requires an API key to use outside of the website.

Ver también:

LibreTranslate website, LibreTranslate repository, LibreTranslate mirrors

2.13.10 Microsoft Terminology

Identificación del servicio
`microsoft-terminology`

Configuración
This service has no configuration.

The Microsoft Terminology Service API allows you to programmatically access the terminology, definitions and user interface (UI) strings available in the Language Portal through a web service.

Ver también:

API del Servicio terminológico de Microsoft

2.13.11 Microsoft Translator

Identificación del servicio
`microsoft-translator`

Configuración

key	Clave de la aplicación	
base_u	La url base de la aplicación	Opciones disponibles: api.cognitive.microsofttranslator.com – Global (non-regional) api-apc.cognitive.microsofttranslator.com – Asia Pacific api-eur.cognitive.microsofttranslator.com – Europe api-nam.cognitive.microsofttranslator.com – North America api.translator.azure.cn – China api.cognitive.microsofttranslator.us – Azure US Government cloud
endpoir	URL del servicio de autenticación	Regional o multi-servicio puede ser especificado usando el campo de region a continuación. Opciones disponibles: api.cognitive.microsoft.com – Global api.cognitive.azure.cn – China api.cognitive.microsoft.us – Azure US Government cloud
region	Región del servicio de autenticación	

Machine translation service provided by Microsoft in Azure portal as a one of Cognitive Services.

Weblate implements Translator API V3.

Translator Text API V2

The key you use with Translator API V2 can be used with API 3.

Translator Text API V3

You need to register at Azure portal and use the key you obtain there. With new Azure keys, you also need to set `region` to locale of your service.

Consejo: For Azure China, please use your endpoint from the Azure Portal.

Ver también:

Cognitive Services - Text Translation API, Microsoft Azure Portal, Base URLs, «Authenticating with a Multi-service resource» «Authenticating with an access token» section

2.13.12 ModernMT

Nuevo en la versión 4.2.

Identificación del servicio
`modernmt`

Configuración

`url`	URL DE LA APLICACIÓN
`key`	Clave de la aplicación

Ver también:

ModernMT API,

2.13.13 MyMemory

Identificación del servicio
`mymemory`

Configuración

`email`	Email de contacto
`username`	Nombre de usuario
`key`	Clave de la aplicación

Enorme memoria de traducción que ofrece traducciones automáticas.

Free, anonymous usage is currently limited to 100 requests/day, or to 1000 requests/day when you provide a contact e-mail address in `email`. You can also ask them for more.

Ver también:

MyMemory website

2.13.14 Netease Sight

Nuevo en la versión 3.3.

Identificación del servicio
`netease-sight`

Configuración

`key`	Identificación del cliente
`secret`	Secreto del cliente

Servicio de traducción automática proporcionado por NetEase.

Este servicio utiliza una API y debe adquirir una clave y secreto de NetEase.

Ver también:

NetEase Sight Translation Platform

2.13.15 SAP Translation Hub

Identificación del servicio
`sap-translation-hub`

Configuración

`url`	URL DE LA APLICACIÓN	
`key`	Clave de la aplicación	
`username`	Nombre de usuario de SAP	
`password`	Contraseña de SAP	
`enabl`	Activar la traducción automática	
`domain`	Dominio de la traducción	El ID de un dominio de traducción, por ejemplo, BC. Si no se especifica un dominio, el método busca las traducciones en todos los dominios disponibles.

Machine translation service provided by SAP.

You need to have a SAP account (and the SAP Translation Hub enabled in the SAP Cloud Platform) to use this service.

You can also configure whether to also use machine translation services, in addition to the term database.

Nota: To access the Sandbox API, you need to set `url` and `key`.

To access the productive API, you need to set `url`, `username` and `password`.

Ver también:

SAP Translation Hub API, Building the Base URL of SAP Translation Hub

2.13.16 tmserver

Identificación del servicio
`tmserver`

Configuración

`url`	URL DE LA APLICACIÓN

You can run your own translation memory server by using the one bundled with Translate-toolkit and let Weblate talk to it. You can also use it with an amaGama server, which is an enhanced version of tmserver.

1. Lo primero que querrá hacer es importar algunos datos en la memoria de traducción:

```
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. Inicie el tmserver para que escuche sus peticiones:

```
tmserver -d /var/lib/tm/db
```

3. Configure Weblate to talk to it, the default URL is `http://localhost:8888/tmserver/`.

Ver también:

tmserver Installing amaGama, Amagama, Amagama Translation Memory

2.13.17 IBM Watson Language Translator

Identificación del servicio
`ibm`

Configuración

`url`	URL DE LA APLICACIÓN
`key`	Clave de la aplicación

IBM Watson Language Translator translates text from one language to another. The service offers multiple domain-specific models.

Ver también:

Watson Language Translator, IBM Cloud API Docs

2.13.18 Weblate

Identificación del servicio
`weblate`

Configuración
This service has no configuration.

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 Translation Memory

Identificación del servicio
`weblate-translation-memory`

Configuración
This service has no configuration.

Use *Memoria de traducción* 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

Identificación del servicio
`yandex`

Configuración

`key`	Clave de la aplicación

Servicio de traducción automática ofrecido por Yandex.

This service uses a Translation API, and you need to obtain an API key from Yandex.

Ver también:

Yandex Translate API, Powered by Yandex.Translate

2.13.21 Youdao Zhiyun

Nuevo en la versión 3.2.

Identificación del servicio
`youdao-zhiyun`

Configuración

`key`	Identificación del cliente
`secret`	Secreto del cliente

El servicio de traducciones automáticas lo brinda Youdao.

Este servicio utiliza una API; habrá de obtener un identificador y una clave de API de Youdao.

Ver también:

Youdao Zhiyun Natural Language Translation Service

2.13.22 Traducción automática personalizada

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):
```

(continué en la próxima página)

(proviene de la página anterior)

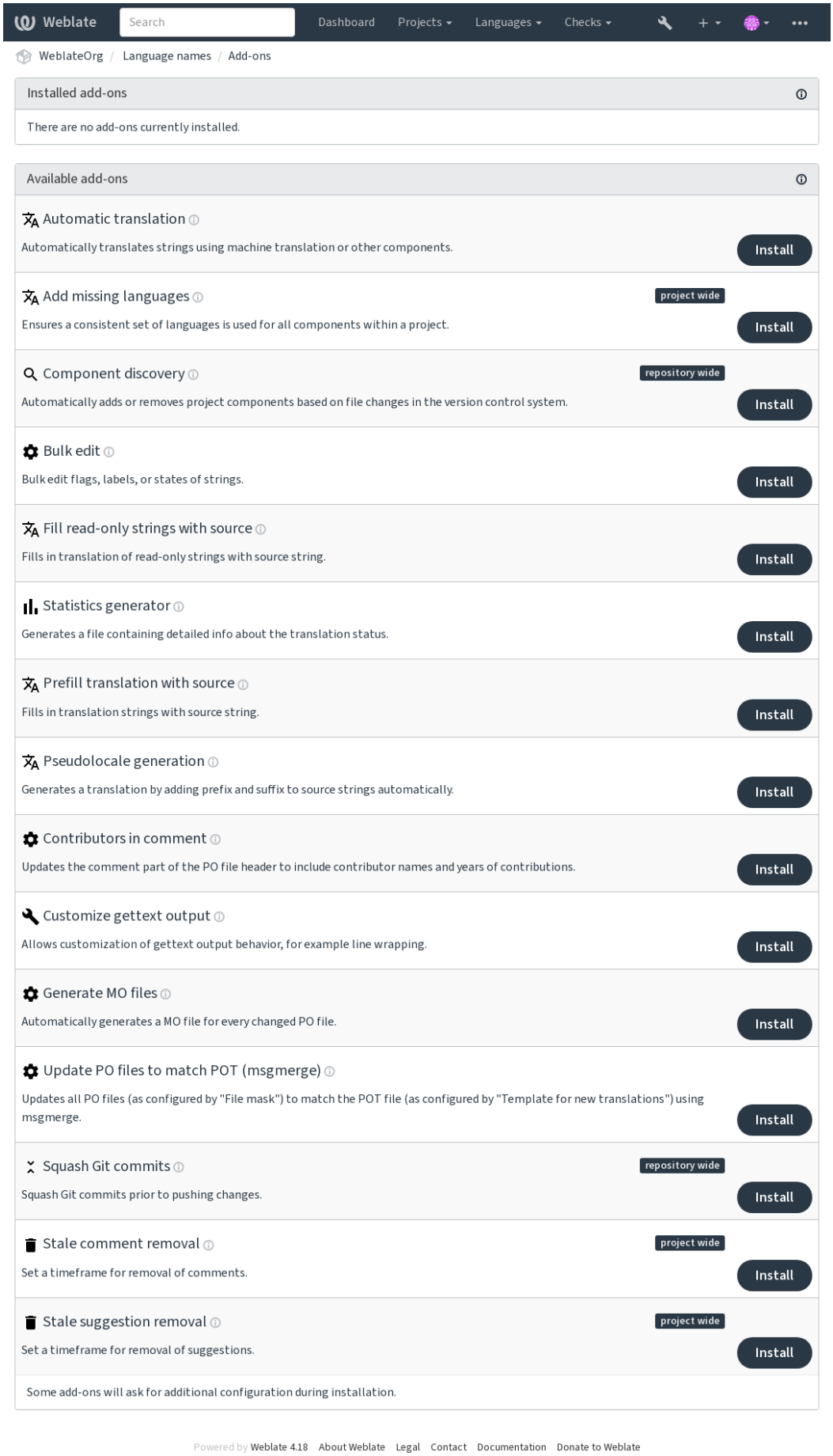
```
        """Return list of languages your machine translation supports."""
        return {"cs"}

    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```

You can list your own class in *WEBLATE_MACHINERY* and Weblate will start using that.

2.14 Complementos

Los complementos proporcionan formas de personalizar y automatizar el flujo de trabajo de la traducción. Los administradores pueden añadir y gestionar complementos desde el menú *Manage* ↓ *Add-ons* de cada componente de la traducción.

Consejo: You can also configure add-ons using *API*, *DEFAULT_ADDONS*, or *weblate weblate install_addon*.


Weblate
Search
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
project wide
Ensures a consistent set of languages is used for all components within a project.
Install
Component discovery
repository wide
Automatically adds or removes project components based on file changes in the version control system.
Install
Bulk edit
Bulk edit flags, labels, or states of strings.
Install
Fill read-only strings with source
Fills in translation of read-only strings with source string.
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
repository wide
Squash Git commits prior to pushing changes.
Install
Stale comment removal
project wide
Set a timeframe for removal of comments.
Install
Stale suggestion removal
project wide
Set a timeframe for removal of suggestions.
Install
Some add-ons will ask for additional configuration during installation.
Powered by Weblate 4.18 About Weblate Legal Contact Documentation Donate to Weblate

2.14.1 Complementos integrados

Traducción automática

Nuevo en la versión 3.9.

ID del complemento
`weblate.traducir automáticamente.traducir automáticamente`

Configuración

`mode`	Modo de traducción automatizada	Opciones disponibles: `suggest` – Añadir como sugerencia `translate` – Añadir como traducción `fuzzy` – Añadir como «Necesita edición»
`filter_`	Filtro de búsqueda	Observe que traducir todas las cadenas descartará todas las traducciones existentes. Opciones disponibles: `all` – Todas las cadenas `nottranslated` – Cadenas no traducidas `todo` – Cadenas inacabadas `fuzzy` – Cadenas marcadas para ser editas `check:inconsistent` – Comprobación fallida: Inconsistente
`auto_s`	Fuente de traducciones automáticas	Opciones disponibles: `others` – Otros componentes de la traducción `mt` – Traducción automática
`component`	Componente	Escriba el slug de un componente para usar lo como origen, manténgalo en blanco para usar todos los componentes del proyecto actual.
`engines`	Motores de traducción automática	
`thresh`	Umbral de puntuación	

Disparadores
actualización de componentes, diaria

Traduce automáticamente las cadenas mediante traducción automática u otros componentes.

Se desencadena:

- Cuando aparecen cadenas nuevas en un componente.
- Una vez al mes por cada componente; esto puede configurarse mediante las *BACKGROUND_TASKS*.

Ver también:

Traducción automática, *Mantener iguales las traducciones entre los componentes*

CDN de regionalización de JavaScript

Nuevo en la versión 4.2.

ID del complemento
`weblate.cdn.cdnjs`

Configuración

`th-reshold`	Umbral de traducción	Umbral para la inclusión de traducciones.
`css_sel`	Selector de CSS	Selector de CSS para detectar elementos regionalizables.
`coo-kie_nam`	Nombre de la «cookie» de idioma	Nombre de la «cookie» que almacena la preferencia de idioma.
`files`	Extraer cadenas desde archivos HTML	Lista de nombres de archivo en el repositorio actual o URL remotos que analizar en busca de cadenas traducibles.

Disparadores
diario, confirmación posterior del repositorio, actualización posterior del repositorio

Publica las traducciones en una red de entrega de contenidos para su uso al regionalizar JavaScript o HTML.

Puede usarse para traducir páginas estáticas HTML o cargar cadenas de localización en el código JavaScript.

Genera una URL única para su componente que pueda incluir para localizarlas en las páginas HTML . Consulte weblate-cdn para más detalles.

Ver también:

cdn-addon-config, weblate-cdn, cdn-addon-extract, cdn-addon-html

Quitar cadenas vacías

Nuevo en la versión 4.4.

ID del complemento
`weblate.cleanup.blank`

Configuración
Este complemento no tiene configuración.

Disparadores
repositorio post-commit, repositorio post-actualización

Quita las cadenas que carecen de traducción de los archivos de traducción.

Use esto para no tener cadenas vacías en los ficheros de traducción (por ejemplo si su biblioteca de localización los marca como faltantes en vez de usar la cadena original).

Ver también:

Does Weblate update translation files besides translations?

Limpieza de archivos de traducción

ID del complemento
`weblate.cleanup.generic`

Configuración
Este complemento no tiene configuración.

Disparadores
repositorio pre-commit, repositorio post-actualización

Actualiza todos los archivos de traducción para que coincidan con el archivo de base monolingüe. Para la mayoría de los formatos de archivo, esto implica quitar las claves de traducción obsoletas que ya no están en el archivo de base.

Ver también:

Does Weblate update translation files besides translations?

Adición de idiomas faltantes

ID del complemento
`weblate.consistencia.idiomas`

Configuración
Este complemento no tiene configuración.

Disparadores
diario, repositorio después de ser añadido

Garantiza que en todos los componentes de un proyecto se utilice el mismo conjunto de idiomas.

Los idiomas que faltan se verifican una vez cada 24 horas y también cuando se agregan nuevos idiomas en Weblate.

A diferencia de la mayoría de los demás complementos, este afecta a todo el proyecto.

Consejo: Traducción automática de cadenas recién agregadas con *Traducción automática*.

Detección de componentes

ID del complemento
`weblate.discovery.discovery`

Configuración

match	Expresión regular utilizada para relacionar con los archivos de traducción	
file_forma	Formato de archivo	
name_templ	Personalizar el nombre del componente	
base_file_	Definir el nombre del archivo de base monolingüe	Dejar vacío para archivos de traducción bilingües.
new_base	Definir el archivo de base para traducciones nuevas	Nombre del archivo utilizado para traducciones nuevas. Para gettext seleccione un archivo .pot.
intermediate_templ	Archivo de idioma intermediario	Nombre del archivo de traducción intermediario. En la mayoría de los casos es un archivo provisto por los desarrolladores y se utiliza al crear las cadenas de origen finales.
language_regex	Filtro de idioma	Expresión regular para filtrar los archivos de traducción cuando se busca una máscara de archivo.
copy_adc	Clonar complementos del componente principal a los nuevos	
remove	Quitar componentes de archivos inexistentes	
confirm	Confirmo que las coincidencias anteriores parecen correctas	

Disparadores
repositorio posterior al ser actualizado

Añade o quita automáticamente componentes del proyecto en función de los cambios de los archivos en el sistema de control de versiones.

La coincidencia se realiza mediante expresiones regulares que permiten una configuración compleja, pero se requieren algunos conocimientos para hacerlo. Se pueden encontrar algunos ejemplos de unos casos comunes en la sección de ayuda del complemento.

The regular expression to match translation files has to contain two named groups to match component and language. All named groups in the regular expression can be used as variables in the template fields.

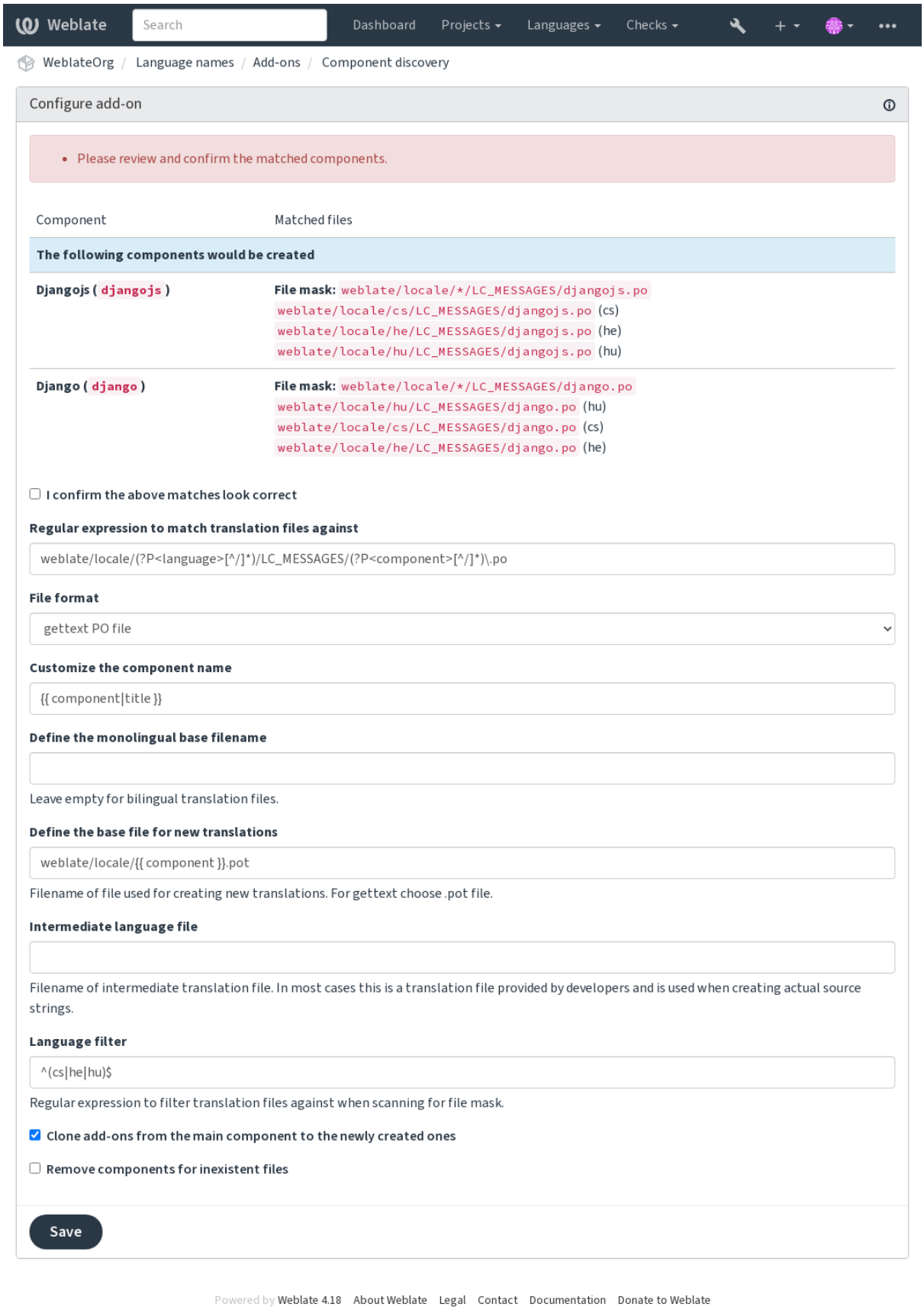
You can use Django template markup in all filename fields, for example:

{{ component }}
Nombre de componente emparejado

{{ component|title }}
Nombre de archivo del componente con la primera letra mayúscula

{{ path }}: {{ component }}
Custom match group from the regular expression

Cuando presiona :guilabel:`Save`aparecerá una vista previa de los componentes coincidentes donde podrá verificar si la configuración se ajusta a sus necesidades:



Component discovery examples

One folder per language

Una carpeta por idioma que contiene los archivos de traducción de los componentes.

Regular expression:

```
(?P<language>[^/.]*)/(?P<component>[^/]*)\.po
```

Matching files:

- `cs/application.po`
- `cs/website.po`
- `de/application.po`
- `de/website.po`

Gettext locales layout

Estructura habitual para almacenar archivos PO de gettext.

Regular expression:

```
locale/(?P<language>[^/.]*)/LC_MESSAGES/(?P<component>[^/]*)\.po
```

Matching files:

- `locale/cs/LC_MESSAGES/application.po`
- `locale/cs/LC_MESSAGES/website.po`
- `locale/de/LC_MESSAGES/application.po`
- `locale/de/LC_MESSAGES/website.po`

Complex filenames

Usando tanto el nombre del componente como el del idioma dentro del nombre del archivo.

Regular expression:

```
src/locale/(?P<component>[^/]*)\.(?P<language>[^/.]*)\.po
```

Matching files:

- `src/locale/application.cs.po`
- `src/locale/website.cs.po`
- `src/locale/application.de.po`
- `src/locale/website.de.po`

Repeated language code

Usar el lenguaje tanto en la ruta como en el nombre del archivo.

Regular expression:

```
locale/(?P<language>[^/.]*)/(?P<component>[^/]*)/(?P=language)\.po
```

Matching files:

- `locale/cs/application/cs.po`
- `locale/cs/website/cs.po`
- `locale/de/application/de.po`
- `locale/de/website/de.po`

Splitted Android strings

Cadenas de recursos de Android, divididas en varios archivos.

Regular expression:

```
res/values-(?P<language>[^/.]*)/strings-(?P<component>[^/]*)\.xml
```

Matching files:

- `res/values-cs/strings-about.xml`
- `res/values-cs/strings-help.xml`
- `res/values-de/strings-about.xml`
- `res/values-de/strings-help.xml`

Matching multiple paths

Proyecto Maven de varios módulos con traducciones de propiedades de Java.

Regular expression:

```
(?P<originalHierarchy>.+/)(?P<component>[^/]*)/src/main/resources/
ApplicationResources_(?P<language>[^/.]*)\.properties
```

Component name:

```
{{ originalHierarchy }}: {{ component }}
```

Matching files:

- `parent/module1/submodule/src/main/resources/ApplicationResources_fr.properties`
- `parent/module1/submodule/src/main/resources/ApplicationResource_es.properties`
- `parent/module2/src/main/resources/ApplicationResource_de.properties`
- `parent/module2/src/main/resources/ApplicationResource_ro.properties`

Consejo: El complemento de detección de los componentes utiliza *URL internos de Weblate*. Es una forma conveniente de compartir la configuración de VCS entre múltiples integrantes. Los componentes vinculados utilizan el repositorio local del componente principal configurado al completar `weblate://project/main-component` en el campo *Repositorio de código fuente* (en *Manage* ↓ :guilabel: *Configuración* ↓ *Sistema de control de versiones*) de cada componente respectivo. Esto también ahorra tiempo con la configuración y los recursos del sistema.

Ver también:

Marcación de plantilla, `weblate weblate import_project`

Edición en masa

Nuevo en la versión 3.11.

ID del complemento
`weblate.flags.bulk`

Configuración

`q`	Consulta	
`state`	Estado que definir	Opciones disponibles: `-1` – No modificar `10` – Necesita ser editado `20` – Traducido `30` – Aprobado
`add_flag`	Indicadores de traducción que añadir	
`remove_flags`	Indicadores de traducción que quitar	
`add_labe`	Etiquetas que añadir	
`remove_label`	Etiquetas que quitar	

Disparadores
actualización de los componentes

Edite los indicadores, las etiquetas o los estados de las cadenas de manera masiva.

Automatice el etiquetado comenzando con la consulta de búsqueda `NO tiene: etiqueta` y agregue etiquetas hasta que todas las cadenas tengan todas las etiquetas requeridas. También se pueden realizar otras operaciones automatizadas para los metadatos de Weblate.

Ejemplos:

Tabla 5: Etiquetar cadenas nuevas automáticamente

Consulta de búsqueda	`NO tiene:etiqueta`
Etiquetas que añadir	*reciente*

Tabla 6: Marcar todas las cadenas del registro de cambios de Archivos de metadatos de tiendas de aplicaciones como de solo lectura

Consulta de búsqueda	`idioma:es Y clave:cambios/`
Indicadores de traducción que añadir	`read-only`

Ver también:

Edición en masa, *Personalizar el comportamiento mediante indicadores*, labels

Marcar traducciones sin cambios como «Necesitan edición»

Nuevo en la versión 3.1.

ID del complemento
`weblate.banderas.mismo_editar`

Configuración
Este complemento no tiene configuración.

Disparadores
unidad posterior a la creación

Cada vez que se importa una cadena traducible nueva desde el VCS y esta corresponde a una cadena de origen, se marca como necesaria la edición en Weblate. Esto es útil sobre todo para aquellos formatos de archivo que incluyen las cadenas de origen de las cadenas no traducidas.

Consejo: También es posible que desee ajustar la verificación *Traducción no modificada* agregando el indicador `strict-same` a *Indicadores de traducción*.

Ver también:

Estados de traducción

Marcar cadenas de origen nuevas como «Necesitan edición»

ID del complemento
`weblate.flags.source_edit`

Configuración
Este complemento no tiene configuración.

Disparadores
unidad posterior a la creación

Cuando se importa una cadena de origen nueva desde el VCS, se marca como necesaria la edición en Weblate. De esta manera usted puede filtrar y editar cadenas de origen escritas por los desarrolladores con facilidad.

Ver también:

Estados de traducción

Marcar traducciones nuevas como «Necesitan edición»

ID del complemento
`weblate.flags.target_edit`

Configuración
Este complemento no tiene configuración.

Disparadores
unidad posterior a la creación

Cuando se importa una cadena traducible nueva desde el VCS, se marca como necesaria la edición en Weblate. De esta manera usted puede filtrar y editar las traducciones creadas por los desarrolladores con facilidad.

Ver también:

Estados de traducción

Fill read-only strings with source

Nuevo en la versión 4.18.

ID del complemento
`weblate.generate.fill_read_only`

Configuración
Este complemento no tiene configuración.

Disparadores
actualización de componentes, diaria

Fills in translation of read-only strings with source string.

Generador de estadísticas

ID del complemento
`weblate.generar.generar`

Configuración

`filename`	Nombre del archivo generado
`template`	Contenido del archivo generado

Disparadores
compromiso previo del repositorio

Genera un archivo con información detallada sobre el estado de la traducción.

Es posible utilizar una plantilla de Django tanto en el nombre de archivo como en el contenido; vea *Marcación de plantilla* para obtener una descripción a detalle de la marcación.

Por ejemplo, para generar un archivo resumen para cada una de las traducciones:

Nombre del archivo generado
`locale/{{ código_de_idioma }}.json`

Contenido

```
{
   "language": "{{ language_code }}",
   "strings": "{{ stats.all }}",
   "translated": "{{ stats.translated }}",
   "last_changed": "{{ stats.last_changed }}",
   "last_author": "{{ stats.last_author }}",
}
```

Ver también:

Marcación de plantilla

Prerrellenar traducción con original

Nuevo en la versión 4.11.

ID del complemento
`weblate.generate.prefill`

Configuración
Este complemento no tiene configuración.

Disparadores
actualización de componentes, diaria

Completa las cadenas de traducción con las de origen.

Todas las cadenas no traducidas del componente se rellenarán con la cadena de origen y se marcarán como: Necesitan edición. Use esto cuando no pueda tener cadenas vacías en los archivos de traducción.

Generación de pseudorregiones

Nuevo en la versión 4.5.

ID del complemento
`weblate.generar.pseudolocalización`

Configuración

`source`	Cadenas de origen	
`target`	Traducción de destino	Se sobrescribirán todas las cadenas de esta traducción
`prefix`	Prefijo de cadena fijo	
`var_prefix`	Prefijo de cadena variable	
`suffix`	Sufijo de cadena fijo	
`var_suffix`	Sufijo de cadena variable	
`var_multip`	Multiplicador de partes variable	En cuántas ocasiones debe repetirse la parte de la variable, en función de la longitud de la cadena original.
`inclu-de_readonl`	Incluir cadenas de solo lectura	

Disparadores
actualización de componentes, diaria

Genera una traducción mediante la adición automática de un prefijo y un sufijo a las cadenas de origen.

Las pseudolocales son útiles para encontrar cadenas que no están preparadas para la localización. Esto se hace alterando todas las cadenas fuente traducibles para facilitar la detección de cadenas no alteradas al ejecutar la aplicación en el idioma pseudolocal.

También es posible encontrar cadenas cuyas contrapartes localizadas no se ajusten al diseño.

El uso de las partes variables hace posible buscar cadenas que podrían no encajar en la interfaz de usuario después de la localización; amplía el texto en función de la longitud de la cadena de origen. Las partes variables se repiten por la longitud del texto multiplicada por el multiplicador. Por ejemplo, `Hola mundo` con el sufijo variable _ y el multiplicador variable de 1 se convierte en `Hola mundo`_____: el sufijo se repite una vez para cada carácter de la cadena de origen.

Las cadenas se generarán acorde a la pauta siguiente:

Prefijo de cadena fija Prefijo de cadena variable Cadena fuente Sufijo de cadena variable Sufijo de cadena fija

Consejo: Puede utilizar idiomas reales para las pruebas, pero hay pseudolocales dedicados disponibles en Weblate : *en_XA* y *ar_XB*.

Consejo: Puede utilizar este complemento para iniciar la traducción a una nueva configuración regional de un idioma existente o un idioma similar. Una vez que agregue la traducción al componente, siga hasta el complemento . *Ejemplo:* Si tiene *fr* y desea iniciar la traducción de *fr_CA* , simplemente establezca *fr* como fuente, *fr_CA* como destino, y deje el prefijo y el sufijo en blanco.

Desinstale el complemento una vez que haya completado la nueva traducción para evitar que Weblate cambie las traducciones realizadas después de la copia.

Contribuidores en comentarios

ID del complemento
`weblate.obtener texto.autores`

Configuración
Este complemento no tiene configuración.

Disparadores
compromiso previo del repositorio

Actualiza el comentario en la cabecera del archivo PO para incluir los nombres de los contribuidores y los años de las contribuciones.

La cabecera del archivo PO lucirá semejante a la siguiente:

```
# 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.
```

Actualizar variable ALL_LINGUAS en el archivo «configure»

ID del complemento
`weblate.obtener texto.configurar`

Configuración
Este complemento no tiene configuración.

Disparadores
repositorio posterior a la adición, diariamente

Actualiza la variable ALL_LINGUAS del archivo `configure`, `configure.in` o cualquier archivo `configure.ac` siempre que se añade una traducción nueva.

Personalizar la salida de gettext

ID del complemento
`weblate.obtener texto.personalizar`

Configuración

`widt`	Ajuste de renglones largos	De manera predeterminada, gettext ajusta los renglones a los 77 caracteres y en los saltos de renglón. Con el parámetro –no-wrap ajusta solo en los saltos. Opciones disponibles: `77` – Envuelve líneas en 77 caracteres y en líneas nuevas (xgettext por defecto) `65535` – Solo ajustar líneas en líneas nuevas (like “xgettext –no-wrap”) `-1` – Sin ajuste de línea

Disparadores
poscarga de almacenamiento

Permite la personalización del comportamiento de salida de gettext, por ejemplo, ajuste de líneas.

Cuenta con las opciones siguientes:

- Ajustar renglones a los 77 caracteres y en saltos de renglón
- Ajustar renglones únicamente en los saltos
- No ajustar renglones

Nota: De manera predeterminada, gettext ajusta los renglones a los 77 caracteres y en los saltos de renglón. Con el parámetro `--no-wrap` ajusta solo en los saltos.

Actualizar archivo LINGUAS

ID del complemento
`weblate.obtener texto.lenguas`

Configuración
Este complemento no tiene configuración.

Disparadores
repositorio posterior a la adición, diariamente

Actualiza el archivo LINGUAS al añadir una traducción nueva.

Generar archivos MO

ID del complemento
`weblate.obtener texto.mo`

Configuración

`path`	Ruta del archivo MO generado	Si no se especifica, se utilizará la ubicación del archivo PO.

Disparadores
compromiso previo del repositorio

Genera automáticamente un archivo MO por cada archivo PO modificado.

La ubicación del archivo MO generado puede personalizarse y el campo correspondiente utiliza *Marcación de plantilla*.

Nota: If a translation is removed, its PO file will be deleted from the repository, but the MO file generated by this add-on will not. The MO file must be removed from the upstream manually.

Actualizar archivos PO para que coincidan con POT (msgmerge)

ID del complemento

`weblate.obtener texto.combinación de mensajes`

Configuración

`previous`	Conservar los msgid anteriores de las cadenas traducidas
`no_location`	Quitar ubicaciones de cadenas traducidas
`fuzzy`	Usar coincidencia aproximada

Disparadores

repositorio posterior al ser actualizado

Actualiza todos los archivos PO (tal como se configuraron mediante *Máscara de archivos*) a juego con el archivo POT (según esté configurado por *Plantilla para traducciones nuevas*) utilizando **`msgmerge`**.

Se activa cada vez que se extraen nuevos cambios del repositorio ascendente. La mayoría de las opciones de línea de comandos de msgmerge se pueden configurar a través de la configuración del complemento.

Ver también:

Does Weblate update translation files besides translations?

Concentrar consignas de Git

ID del complemento

`weblate.git.squash`

Configuración

`squash`	Concentración de consignas	Opciones disponibles: `all` – Todas las confirmaciones en una `idioma` – Por idioma `archivo` – Por archivo `autor` – Por autor
`append_t`	Añadir renglones finales al mensaje de consigna concentrada	Las líneas de acompañamiento son líneas similares a las cabeceras de correo electrónico RFC 822, al final de la parte libre de un mensaje de confirmación, como «Coautor: …».
`commit_me`	Mensaje de consigna	Este mensaje de consigna se utilizará en sustitución de los mensajes combinados de las consignas concentradas.

Disparadores

repositorio posterior a ser confirmado

Concentrar las consignas de Git antes de enviar los cambios.

Las consignas Git se pueden concentrar antes de enviar los cambios en uno de los modos siguientes:

- Todas las consignas en una sola
- Por idioma
- Por archivo
- Por autor

Se conservarán los mensajes de consigna originales pero la autoría se perderá, a menos que seleccione *Por autor* o personalice el mensaje de consigna para incluirla.

Los mensajes de confirmación originales se pueden anular opcionalmente con un mensaje de confirmación personalizado.

Existe la opción de quitar los renglones finales (como `Co-authored-by: …`) de los mensajes de consigna originales y reañadirlos al mensaje de la consigna concentrada. `Co-authored-by:` asegurará además la debida acreditación a cada uno de los traductores.

Personalizar la salida de JSON

ID del complemento
`weblate.json.personalizado`

Configuración

`sort_key`	Ordenar claves de JSON	
`indent`	Sangría JSON	
`style`	Estilo de sangría de JSON	Opciones disponibles: `espacios` – Espacios `pestañas` – Pestañas

Disparadores
poscarga de almacenamiento

Permite ajustar el comportamiento de salida de JSON, por ejemplo, sangría o clasificación.

Formatear el archivo de propiedades de Java

ID del complemento
`weblate.propiedades.ordenar`

Configuración
Este complemento no tiene configuración.

Disparadores
compromiso previo del repositorio

Formatea y ordena el archivo de propiedades de Java.

- Consolida nuevas líneas a las de Unix.
- Formateo en mayúsculas de secuencias de escape Unicode (en caso de que estén presentes).
- Elimina las líneas en blanco y los comentarios.
- Ordena las cadenas por las claves.
- Deja las cadenas duplicadas.

Retiro de comentarios obsoletos

Nuevo en la versión 3.7.

ID del complemento
`weblate.eliminación.comentarios`

Configuración

`age`	Días para guardar

Disparadores
Diario

Fijar un plazo para el borrado de los comentarios.

Esto puede resultar útil para quitar los comentarios antiguos que pueden ya no ser aplicables. Utilícela con cuidado, pues el hecho de que un comentario sea antiguo no necesariamente denota que ha perdido importancia.

Retiro de sugerencias obsoletas

Nuevo en la versión 3.7.

ID del complemento
`weblate.eliminación.sugerencias`

Configuración

`age`	Días para guardar	
`votes`	Umbral de votos	Umbral para eliminación. Este campo no surte ningún efecto si se desactivan los votos.

Disparadores
Diario

Fijar un plazo para el borrado de sugerencias.

Puede ser muy útil en relación con la votación de sugerencias (ver *Revisión por pares*) para eliminar sugerencias que no reciben suficientes votos positivos en un período de tiempo determinado.

Actualizar archivos RESX

Nuevo en la versión 3.9.

ID del complemento
`weblate.resx.actualizar`

Configuración
Este complemento no tiene configuración.

Disparadores
repositorio posterior al ser actualizado

Actualiza todos los archivos de traducción para que coincidan con el archivo de base monolingüe original. Las cadenas no utilizadas se retiran y las nuevas se añaden como copias de las cadenas de origen.

Consejo: Use *Limpieza de archivos de traducción* si solo desea eliminar las claves de traducción obsoletas.

Ver también:

Does Weblate update translation files besides translations?

Personalizar la salida XML

Nuevo en la versión 4.15.

ID del complemento
`weblate.xml.customize`

Configuración

`closing_tags`	Incluir etiqueta de cierre para etiquetas XML vacías

Disparadores
poscarga de almacenamiento

Allows adjusting XML output behavior, for example closing tags.

Personalizar salida de YAML

Nuevo en la versión 3.10.2.

ID del complemento
`weblate.yaml.personalizar`

Configuración

`indent`	Sangría YAML	
`width`	Ajuste de renglones largos	Opciones disponibles: `80` – Envuelve líneas en 80 caracteres `100` – Envuelve líneas en 100 caracteres `120` – Envuelve líneas en 120 caracteres `180` – Envuelve líneas en 180 caracteres `65535` – Sin ajuste de línea
`line_break`	Saltos de renglón	Opciones disponibles: `dos` – DOS (\r\n) `unix` – UNIX (\n) `mac` – MAC (\r)

Disparadores
poscarga de almacenamiento

Permite ajustar el comportamiento de salida de YAML, como la longitud de los renglones y los saltos.

2.14.2 Personalización de la lista de complementos

La lista de complementos se configura mediante *WEBLATE_ADDONS*. Para agregar otro complemento, simplemente incluya el nombre absoluto de la clase en esta configuración.

2.14.3 Complemento de escritura

También puede escribir sus propios complementos, crear una subclase de *weblate.addons.base.BaseAddon* para definir los metadatos del complemento y luego implementar una devolución de llamada para realizar el procesamiento.

Ver también:

Developing add-ons

2.14.4 Ejecutar scripts desde el complemento

Los add-ons pueden utilizarse para ejecutar scripts externos. Esto solía estar integrado en Weblate, pero ahora tienes que escribir algo de código para envolver tu script con un add-on.

```
# 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

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 = gettext_lazy("Execute script before commit")
    description = gettext_lazy("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"
```

Para obtener instrucciones de instalación, vea *Controles de calidad personalizados, complementos y autocorrecciones*.

El script se ejecuta con el directorio actual establecido en la raíz del repositorio VCS para cualquier componente dado.

Además, las variables de entorno siguientes están a su disposición:

WL_VCS
: Sistema de control de versiones utilizado.

WL_REPO
: URL del repositorio del proyecto originario.

WL_PATH

Ruta absoluta al repositorio del sistema de control de versiones.

WL_BRANCH

Rama del repositorio configurada en el componente actual.

WL_FILEMASK

Máscara de archivos para el componente actual.

WL_TEMPLATE

Nombre de archivo de plantilla para traducciones monolingües (puede quedar vacío).

WL_NEW_BASE

Nombre del archivo utilizado para crear traducciones nuevas (puede quedar vacío).

WL_FILE_FORMAT

Formato de archivo utilizado en el componente actual.

WL_LANGUAGE

Idioma de la traducción procesada actualmente (no disponible para actuadores a nivel componente).

WL_PREVIOUS_HEAD

HEAD anterior después de la actualización (sólo disponible después de ejecutar el hook posterior a la actualización).

WL_COMPONENT_SLUG

Nuevo en la versión 3.9.

Componente slug utilizado para construir la URL.

WL_PROJECT_SLUG

Nuevo en la versión 3.9.

Slug del proyecto utilizado para construir la URL.

WL_COMPONENT_NAME

Nuevo en la versión 3.9.

Nombre del componente.

WL_PROJECT_NAME

Nuevo en la versión 3.9.

Nombre del proyecto.

WL_COMPONENT_URL

Nuevo en la versión 3.9.

URL del componente.

WL_ENGAGE_URL

Nuevo en la versión 3.9.

URL de participación del proyecto.

Ver también:

Configuración de componentes

Procesando el repositorio después de la actualización

Se puede utilizar para actualizar los archivos de traducción cuando el código fuente del VCS cambia. Para conseguirlo, recuerda que Weblate sólo ve los archivos confirmados en el VCS, por lo que debes commit los cambios como parte del script.

Por ejemplo, con Gulp puede hacerlo a través del código a continuación:

```
#! /bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Procesamiento de traducciones preconsigna

Utilice el script commitn para cambiar automáticamente una traducción antes de confirmarla en el repositorio.

Se pasa como un único parámetro que consiste en el nombre de archivo de una traducción actual.

2.15 Memoria de traducción

Weblate incluye una memoria de traducción incorporada que consiste de lo siguiente:

- Manually imported translation memory (see *Interfaz de usuario*).
- Automatically stored translations performed in Weblate (depending on *Ámbitos de la memoria de traducción*).
- Traducciones pasadas importadas automáticamente.

El contenido de la memoria de traducción puede aplicarse de dos maneras:

- Manually, *Sugerencias automáticas* view while translating.
- Automatically, by translating strings using *Traducción automática*, or *Traducción automática* add-on.

For installation tips, see *Weblate Translation Memory*, which is turned on by default.

2.15.1 Ámbitos de la memoria de traducción

Nuevo en la versión 3.2: In earlier versions translation memory could be only loaded from a file corresponding to the current imported translation memory scope.

The translation memory scopes are there to allow both privacy and sharing of translations, to suit the desired behavior.

Memoria de traducción importada

Importing arbitrary translation memory data using the `weblate weblate import_memory` command makes memory content available to all users and projects.

Memoria de traducción por usuario

Stores all user translations automatically in the personal translation memory of each respective user.

Memoria de traducción por proyecto

All translations within a project are automatically stored in a project translation memory only available for this project.

Memoria de traducción compartida

Todas las traducciones que se realicen en proyectos con la memoria compartida activada se almacenan de modo tal que quedarán disponibles para todos los proyectos.

Pondere si quiere activar esta funcionalidad en instalaciones compartidas de Weblate, ya que puede tener implicaciones importantes:

- Cualquiera puede hacer uso de las traducciones.
- Esto podría causar la revelación de información secreta.

Ver también:

Contribuir a la memoria de traducción compartida, *Utilizar memoria de traducción compartida*

2.15.2 Gestionar la memoria de traducción

Interfaz de usuario

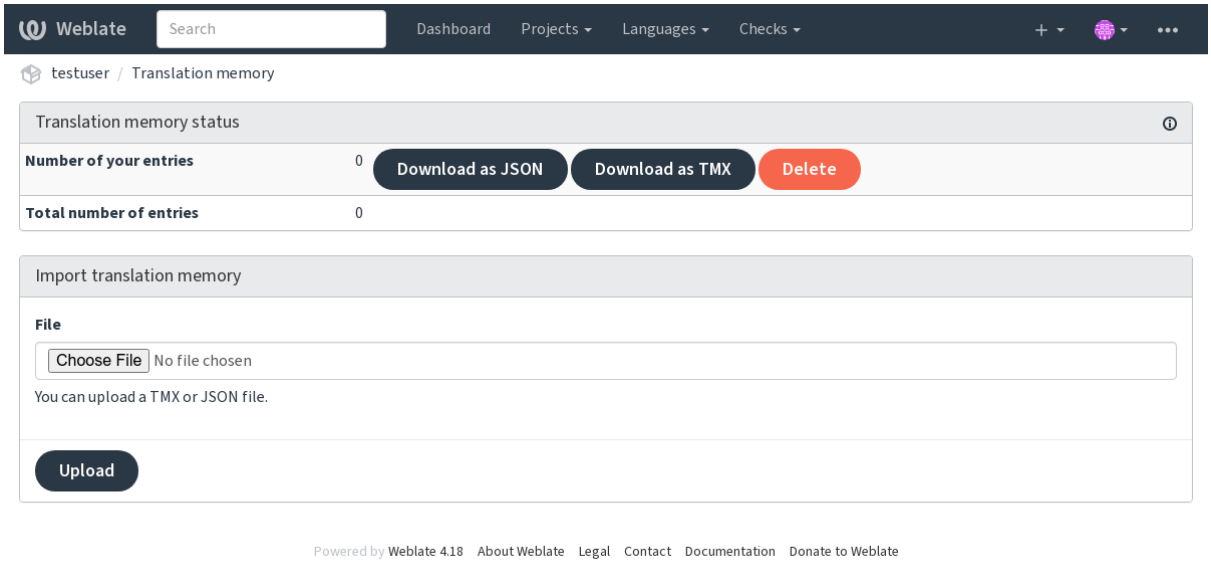
Nuevo en la versión 3.2.

In the basic user interface you can manage per user and per project translation memories. It can be used to download, wipe or import translation memory.

Consejo: Las memorias de traducción JSON pueden importarse en Weblate. Se brinda el formato TMX por razones de interoperatividad con otras herramientas.

Ver también:

Weblate Translation Memory Schema



Interfaz de gestión

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):

weblate weblate dump_memory
Exporta la memoria a JSON

weblate weblate import_memory
Importa archivos TMX o JSON en la memoria de traducción

2.16 Configuración

Todos los ajustes se almacenan en `settings.py` (como es habitual en Django).

Nota: After changing any of these settings, you need to restart Weblate — both WSGI and Celery processes.

En caso de que se ejecute como `mod_wsgi`, es necesario reiniciar Apache para recargar la configuración.

Ver también:

Por favor, consulta también La documentación de Django para conocer los parámetros de configuración del propio Django.

2.16.1 AKISMET_API_KEY

Weblate puede utilizar Akismet para comprobar las sugerencias anónimas entrantes en busca de spam. Visita akismet.com para comprar una clave API y asociarla a un sitio.

2.16.2 ANONYMOUS_USER_NAME

El nombre de usuario de los usuarios sin cuenta.

Ver también:

Control de acceso

2.16.3 AUDITLOG_EXPIRY

Nuevo en la versión 3.6.

How many days Weblate should keep audit logs (which contain info about account activity).

El valor predeterminado es de 180 días.

2.16.4 AUTH_LOCK_ATTEMPTS

Número máximo de intentos de autenticación fallidos antes de que se aplique la limitación de velocidad.

Actualmente se aplica en los siguientes lugares:

- Iniciar sesión. Elimina la contraseña de la cuenta, impidiendo que el usuario se registre sin solicitar una nueva contraseña.
- Password reset. Prevents new e-mails from being sent, avoiding spamming users with too many password-reset attempts.

El valor predeterminado es de 10.

Ver también:

Rate limiting

2.16.5 AUTO_UPDATE

Nuevo en la versión 3.2.

Distinto en la versión 3.11: La opción original de activación/desactivación se modificó para diferenciar las cadenas que se aceptan.

Actualiza todos los repositorios diariamente.

Consejo: Útil si no está utilizando *Actuadores de notificación* para actualizar los repositorios de Weblate automáticamente.

Nota: Existen opciones de activación/desactivación además de la selección de cadenas para la compatibilidad con versiones anteriores.

The options are:

`"none"`
: No hay actualizaciones diarias.

`"remote"` también `False`
: Actualiza los remotos solamente.

`"full"` también `True`
: Actualiza los remotos y fusiona la copia de trabajo.

Nota: Esto requiere que *Tareas en segundo plano con Celery* esté funcionando, y tendrá efecto después de que se reinicie.

2.16.6 AVATAR_URL_PREFIX

Prefijo para construir las URL de los avatares como: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Se sabe que los siguientes servicios funcionan:

Gravatar (por defecto), según https://gravatar.com/

```
AVATAR_URL_PREFIX = 'https://www.gravatar.com/'
```

Libravatar, según https://www.libravatar.org/

```
AVATAR_URL_PREFIX = 'https://www.libravatar.org/'
```

Ver también:

Caché de avatars, *ENABLE_AVATARS*, *Avatars*

2.16.7 AUTH_TOKEN_VALID

El tiempo de validez del token de autenticación y la contraseña temporal de los correos electrónicos de restablecimiento de contraseña. Se establece en número de segundos, por defecto 172800 (2 días).

2.16.8 AUTH_PASSWORD_DAYS

How many days Weblate rejects reusing a previously used password for a user.

The checking is based on the audit log, *AUDITLOG_EXPIRY* needs to be at least same as this.

Nota: Los cambios de contraseña realizados antes de Weblate 2.15 no se tendrán en cuenta en esta política.

El valor predeterminado es de 180 días.

2.16.9 AUTOFIX_LIST

Lista de correcciones automáticas que aplicar al guardar una cadena.

Nota: Provide a fully-qualified path to the Python class that implements the autofixer interface.

Correcciones disponibles:

weblate.trans.autofixes.whitespace.SameBookendingWhitespace
: Hacer coincidir los espacios en blanco al principio y al final de la cadena con la fuente.

weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis
: Sustituye tres puntos al final (...) si la cadena de origen incluye en su lugar puntos suspensivos (…).

weblate.trans.autofixes.chars.RemoveZeroSpace
: Quita caracteres espaciadores de anchura cero si la cadena de origen no los contiene.

weblate.trans.autofixes.chars.RemoveControlChars
: Quita caracteres de control si la cadena de origen no los contiene.

weblate.trans.autofixes.chars.DevanagariDanda
: Replaces sentence full stop in Bangla by the devanagari danda character.

weblate.trans.autofixes.html.BleachHTML
: Elimina las marcas HTML no seguras de las cadenas marcadas como `safe-html` (ver *HTML inseguro*).

Puede seleccionar cuáles utilizar:

```
AUTOFIX_LIST = (
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
)
```

Ver también:

Correcciones automáticas, *Correcciones automáticas personalizadas*

2.16.10 BACKGROUND_TASKS

Nuevo en la versión 4.5.2.

Define la frecuencia con la que deben activarse las tareas de mantenimiento prolongadas para un componente.

En este momento esto controla:

- *Traducción automática* add-on
- *Comprobaciones y correcciones* recalcular

Elecciones posibles:

- `monthly` (valor predeterminado)
- `weekly`
- `daily`
- `never`

Nota: No es recomendable aumentar la frecuencia cuando Weblate contiene miles de componentes.

2.16.11 BASIC_LANGUAGES

Nuevo en la versión 4.4.

List of languages to offer users for starting a new translation. When not specified, a built-in list is used (which includes all commonly used languages, but without country specific variants).

This only limits non privileged users to add unwanted languages. Project admins are still presented with the full selection of languages defined in Weblate.

Nota: This does not define new languages for Weblate — it only filters existing ones in the database.

Ejemplo:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Ver también:

Definiciones de idioma

2.16.12 BORG_EXTRA_ARGS

Nuevo en la versión 4.9.

Puedes pasar argumentos adicionales a **`borg create`** cuando se activan las copias de seguridad integradas.

Ejemplo:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Ver también:

Respaldar y trasladar Weblate, borg create

2.16.13 CACHE_DIR

Nuevo en la versión 4.16.

Directory where Weblate stores cache files. Defaults to `cache` subfolder in *DATA_DIR*.

Change this to local or temporary filesystem if *DATA_DIR* is on a network filesystem.

El contenedor Docker utiliza un volumen separado para esto, ver *Volúmenes de contenedores Docker*.

2.16.14 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Personaliza la cabecera `Content-Security-Policy` para Weblate. La cabecera se genera automáticamente en función de las integraciones habilitadas con servicios de terceros (Matomo, Google Analytics, Sentry, …).

Todo esto es una lista vacía por defecto.

Ejemplo:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Ver también:

Content security policy, Directiva de seguridad de contenido (CSP)

2.16.15 CHECK_LIST

Lista de comprobaciones de calidad que ejecutar en una traducción.

Nota: Proporcione una ruta completa a la clase de Python que implementa la interfaz para la comprobación.

Ajuste la lista de comprobaciones para incluir aquellas que le sean relevantes.

All built-in *Comprobaciones de calidad* are turned on by default, from where you can change these settings. By default they are commented out in *Configuración de muestra* so that default values are used. New checks are then carried out for each new Weblate version.

Puede desactivar todas las comprobaciones:

```
CHECK_LIST = ()
```

Puede activar solo algunas:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Nota: Changing this setting only affects newly changed translations. Existing checks will still be stored in the database. To also apply changes to the stored translations, run `weblate weblate updatechecks`.

Ver también:

Comprobaciones de calidad, *Personalizar el comportamiento mediante indicadores*

2.16.16 COMMENT_CLEANUP_DAYS

Nuevo en la versión 3.6.

Eliminar los comentarios después de un número determinado de días. El valor predeterminado es `None`, lo que significa que no se borra nada.

2.16.17 COMMIT_PENDING_HOURS

Número de horas entre la confirmación de los cambios pendientes mediante la tarea en segundo plano.

Ver también:

Configuración de componentes, *Antigüedad de cambios por consignar*, *Efectuar tareas de mantenimiento*, `weblate weblate commit_pending`

2.16.18 CONTACT_FORM

Nuevo en la versión 4.6.

Configures how e-mail from the contact form is being sent. Choose a configuration that matches the configuration of your mail server.

"reply-to"
: El remitente se utiliza como *Reply-To*, este es el comportamiento por defecto.

"from"
: El remitente se utiliza como *From*. Su servidor de correo debe permitir el envío de este tipo de correos.

2.16.19 DATA_DIR

La carpeta en la que Weblate almacena todos los datos. Contiene enlaces a los repositorios VCS, un índice de texto completo y varios archivos de la configuración para herramientas externas.

A menudo existen los subdirectorios siguientes:

home
: Directorio de inicio utilizado para invocar secuencias de órdenes.

ssh
: Claves SSH y su configuración.

static
: Ubicación por defecto para los archivos estáticos de Django, especificada por `STATIC_ROOT`. Ver *Serving static files*.

  El contenedor Docker utiliza un volumen separado para esto, ver *Volúmenes de contenedores Docker*.

media
: Ubicación por defecto para los archivos multimedia de Django, especificada por `MEDIA_ROOT`. Contiene las capturas de pantalla subidas, véase *Contexto visual para cadenas*.

vcs
: Version-control repositories for translations.

backups
: Daily backup data. Please check *Datos volcados para las copias de respaldo* for details.

fonts:
: Fuentes cargadas por el usuario, véase *Gestionar tipos de letra*.

cache
: Various caches. Can be placed elsewhere using *CACHE_DIR*.

  El contenedor Docker utiliza un volumen separado para esto, ver *Volúmenes de contenedores Docker*.

Nota: Weblate necesita ser capaz de escribir en este directorio. Ejecutarlo como uWSGI implica que el usuario `www-data` debe tener privilegios de escritura aquí.

La manera más sencilla de lograrlo es hacer que el usuario sea propietario del directorio:

```
sudo chown www-data:www-data -R $DATA_DIR
```

Por defecto es `/home/weblate/data`, pero se espera que esté configurado.

Ver también:

Permisos del sistema de archivos, *Respaldar y trasladar Weblate*, *CACHE_DIR*

2.16.20 DATABASE_BACKUP

Nuevo en la versión 3.1.

Define si las copias de respaldo de la base de datos deben almacenarse como texto sencillo, comprimidas u omitirse. Los valores autorizados son:

- `"plain"`
- `"compressed"`
- `"none"`

Ver también:

Respaldar y trasladar Weblate

2.16.21 DEFAULT_ACCESS_CONTROL

Nuevo en la versión 3.3.

The default access-control setting for new projects:

0
: *Público*

1
: *Protegido*

100
: *Privado*

200
: *Personalizado*

Utilice *Custom* si está gestionando ACL manualmente, lo que significa no depender de la gestión interna de Weblate.

Ver también:

Control de acceso al proyecto, *Control de acceso*

2.16.22 DEFAULT_AUTO_WATCH

Nuevo en la versión 4.5.

Configura si la etiqueta *Automatically watch projects on contribution* debe ser activada para los nuevos usuarios. Por defecto es `True`.

Ver también:

Notificaciones

2.16.23 DEFAULT_RESTRICTED_COMPONENT

Nuevo en la versión 4.1.

El valor predeterminado para la restricción de componentes.

Ver también:

Acceso restringido, *Ámbito de los equipos*

2.16.24 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DEFAULT_MERGE_MESSAGE

Mensajes de confirmación por defecto para diferentes operaciones, por favor revise componente para más detalles.

Ver también:

Marcación de plantilla, *Configuración de componentes*, *Confirmar, agregar, eliminar, fusionar, agregar y fusionar mensajes de la solicitud*

2.16.25 DEFAULT_ADDONS

Default add-ons to install for every created component.

Nota: Esta configuración solo surte efecto en los componentes nuevos.

Ejemplo:

```
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",
    },
}
```

Ver también:

`weblate weblate install_addon`, *Complementos*, `WEBLATE_ADDONS`

2.16.26 DEFAULT_COMMITER_EMAIL

Committer e-mail address, defaulting to `noreply@weblate.org`.

Ver también:

DEFAULT_COMMITER_NAME

2.16.27 DEFAULT_COMMITER_NAME

Committer name, defaulting to `Weblate`.

Ver también:

DEFAULT_COMMITER_EMAIL

2.16.28 DEFAULT_LANGUAGE

Nuevo en la versión 4.3.2.

Idioma del código fuente for any new components.

El valor predeterminado es *en*. El objeto de lenguaje correspondiente debe existir en la base de datos.

Ver también:

Definiciones de idioma, *Idioma del código fuente*

2.16.29 DEFAULT_MERGE_STYLE

Nuevo en la versión 3.4.

Estilo de fusión for any new components.

- *rebase* - por defecto
- *merge*

Ver también:

Configuración de componentes, *Estilo de fusión*

2.16.30 DEFAULT_SHARED_TM

Nuevo en la versión 3.2.

Configures the default value of *Utilizar memoria de traducción compartida* and *Contribuir a la memoria de traducción compartida*.

2.16.31 DEFAULT_TRANSLATION_PROPAGATION

Configuración por defecto para la propagación de la traducción, por defecto es `True`.

Ver también:

Configuración de componentes, *Permitir propagación de traducciones*

2.16.32 DEFAULT_PULL_MESSAGE

Configura el título y el mensaje predeterminados para las solicitudes de extracción.

2.16.33 ENABLE_AVATARS

Whether to turn on Gravatar-based avatars for users. On by default.

Los avatares se obtienen y se almacenan en caché en el servidor, lo que reduce el riesgo de que se filtre información privada y acelera la experiencia del usuario.

Ver también:

Caché de avatars, `AVATAR_URL_PREFIX`, *Avatars*

2.16.34 ENABLE_HOOKS

Whether to turn on anonymous remote hooks.

Ver también:

Actuadores de notificación

2.16.35 ENABLE_HTTPS

Si se envían los enlaces a Weblate como HTTPS o HTTP. Esta configuración afecta a los correos electrónicos enviados y a las URL absolutas generadas.

In the default configuration this is also used for several Django settings related to HTTPS — it enables secure cookies, toggles HSTS or enables redirection to a HTTPS URL.

The HTTPS redirection might be problematic in some cases and you might hit an issue with infinite redirection in case you are using a reverse proxy doing an SSL termination which does not correctly pass protocol headers to Django. Please tweak your reverse proxy configuration to emit `X-Forwarded-Proto` or `Forwarded` headers or configure `SECURE_PROXY_SSL_HEADER` to let Django correctly detect the SSL status.

Ver también:

`SESSION_COOKIE_SECURE`, `CSRF_COOKIE_SECURE`, `SECURE_SSL_REDIRECT`, `SECURE_PROXY_SSL_HEADER` *Set correct site domain*

2.16.36 ENABLE_SHARING

Activa/desactiva el menú *Compartir* para que los usuarios puedan compartir el progreso de la traducción en las redes sociales.

2.16.37 EXTRA_HTML_HEAD

Nuevo en la versión 4.15.

Insert additional markup into the HTML header. Can be used for verification of site ownership, for example:

```
EXTRA_HTML_HEAD = '<link href="https://fosstodon.org/@weblate" rel="me">'
```

Advertencia: No sanitization is performed on the string. It is inserted as-is into the HTML header.

2.16.38 GET_HELP_URL

Nuevo en la versión 4.5.2.

URL en el que pueden encontrarse recursos de ayuda para su instalación de Weblate.

2.16.39 GITEA_CREDENTIALS

Nuevo en la versión 4.12.

Lista de credenciales para los servidores de Gitea.

```
GITEA_CREDENTIALS = {
    "try.gitea.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
}
```

The configuration dictionary consists of credentials defined for each API host. The API host might be different from what you use in the web browser, for example GitHub API is accessed as `api.github.com`.

The following configuration is available for each host:

username
API user, required.

token
API token for the API user, required.

scheme
Nuevo en la versión 4.18.

Scheme override. Weblate attempts to parse scheme from the repository URL and falls backs to `https`. If you are running the API server internally, you might want to use `http` instead, but consider security.

Consejo: In the Docker container, the credentials are configured in three variables and the credentials are built out of that. An example configuration for GitHub might look like:

```
WEBLATE_GITHUB_USERNAME=api-user
WEBLATE_GITHUB_TOKEN=api-token
WEBLATE_GITHUB_HOST=api.github.com
```

Will be used as:

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "api-user",
        "token": "api-token",
    }
}
```

Ver también:

Solicitudes de incorporación de Gitea, Creating a Gitea personal access token

2.16.40 GITLAB_CREDENTIALS

Nuevo en la versión 4.3.

Lista de credenciales para los servidores de GitLab.

```
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
}
```

The configuration dictionary consists of credentials defined for each API host. The API host might be different from what you use in the web browser, for example GitHub API is accessed as `api.github.com`.

The following configuration is available for each host:

username
API user, required.

token
API token for the API user, required.

scheme
Nuevo en la versión 4.18.

Scheme override. Weblate attempts to parse scheme from the repository URL and falls backs to `https`. If you are running the API server internally, you might want to use `http` instead, but consider security.

Consejo: In the Docker container, the credentials are configured in three variables and the credentials are built out of that. An example configuration for GitHub might look like:

```
WEBLATE_GITHUB_USERNAME=api-user
WEBLATE_GITHUB_TOKEN=api-token
WEBLATE_GITHUB_HOST=api.github.com
```

Will be used as:

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "api-user",
        "token": "api-token",
    }
}
```

Ver también:

Solicitudes de fusión de GitLab, GitLab: Personal access token

2.16.41 GITHUB_CREDENTIALS

Nuevo en la versión 4.3.

Lista de credenciales para los servidores de GitHub.

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
}
```

The configuration dictionary consists of credentials defined for each API host. The API host might be different from what you use in the web browser, for example GitHub API is accessed as `api.github.com`.

The following configuration is available for each host:

username
API user, required.

token
API token for the API user, required.

scheme
Nuevo en la versión 4.18.

Scheme override. Weblate attempts to parse scheme from the repository URL and falls backs to `https`. If you are running the API server internally, you might want to use `http` instead, but consider security.

Consejo: In the Docker container, the credentials are configured in three variables and the credentials are built out of that. An example configuration for GitHub might look like:

```
WEBLATE_GITHUB_USERNAME=api-user
WEBLATE_GITHUB_TOKEN=api-token
WEBLATE_GITHUB_HOST=api.github.com
```

Will be used as:

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "api-user",
        "token": "api-token",
    }
}
```

Ver también:

Solicitudes de incorporación de GitHub, Creating a GitHub personal access token

2.16.42 BITBUCKETSERVER_CREDENTIALS

Nuevo en la versión 4.16.

List for credentials for Bitbucket servers.

```
BITBUCKETSERVER_CREDENTIALS = {
    "git.self-hosted.com": {
        "username": "weblate",
        "token": "http-access-token",
    },
}
```

The configuration dictionary consists of credentials defined for each API host. The API host might be different from what you use in the web browser, for example GitHub API is accessed as `api.github.com`.

The following configuration is available for each host:

username
API user, required.

token
API token for the API user, required.

scheme
Nuevo en la versión 4.18.

Scheme override. Weblate attempts to parse scheme from the repository URL and falls backs to `https`. If you are running the API server internally, you might want to use `http` instead, but consider security.

Consejo: In the Docker container, the credentials are configured in three variables and the credentials are built out of that. An example configuration for GitHub might look like:

```
WEBLATE_GITHUB_USERNAME=api-user
WEBLATE_GITHUB_TOKEN=api-token
WEBLATE_GITHUB_HOST=api.github.com
```

Will be used as:

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "api-user",
        "token": "api-token",
    }
}
```

Ver también:

Bitbucket Server pull requests, Bitbucket: HTTP access token

2.16.43 GOOGLE_ANALYTICS_ID

ID de Google Analytics para activar el seguimiento de Weblate mediante Google Analytics.

2.16.44 HIDE_REPO_CREDENTIALS

Ocultar las credenciales del repositorio en la interfaz web. En caso de tener la URL del repositorio con el usuario y la contraseña, Weblate la ocultará cuando se muestre la información relacionada a los usuarios.

Por ejemplo, en lugar de `https://user:password@git.example.com/repo.git` mostrará sólo `https://git.example.com/repo.git`. También intenta limpiar los mensajes de los errores de VCS de forma similar.

Nota: On by default.

2.16.45 HIDE_VERSION

Nuevo en la versión 4.3.1.

Hides version info from unauthenticated users. This also makes all documentation links point to the latest version instead of the documentation matching the currently installed version.

Hiding the version is a recommended security practice in some corporations, does not prevent an attacker from figuring out version by probing behavior.

Nota: Está desactivado por defecto.

2.16.46 INTERLEDGER_PAYMENT_POINTERS

Nuevo en la versión 4.12.1.

Lista de indicadores de pago interledger (ILP) para a monetización web.

Si se especifican varios, el reparto probabilístico de los ingresos se consigue seleccionando uno al azar.

Consulte <https://webmonetization.org/> para obtener más detalles.

Consejo: The default value lets users fund Weblate itself.

2.16.47 IP_BEHIND_REVERSE_PROXY

Indica si Debiste se está ejecutando detrás de un proxy reverso.

If set to `True`, Weblate gets IP address from a header defined by *IP_PROXY_HEADER*.

Advertencia: Ensure you are actually using a reverse proxy and that it sets this header, otherwise users will be able to fake the IP address.

Nota: No está activado por defecto.

Ver también:

Running behind reverse proxy, *Rate limiting*, *IP_PROXY_HEADER*, *IP_PROXY_OFFSET*

2.16.48 IP_PROXY_HEADER

Indicates which header Weblate should obtain the IP address from when *IP_BEHIND_REVERSE_PROXY* is turned on.

Defaults to `HTTP_X_FORWARDED_FOR`.

Ver también:

Running behind reverse proxy, *Rate limiting*, `SECURE_PROXY_SSL_HEADER`, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_OFFSET*

2.16.49 IP_PROXY_OFFSET

Indicates which part of *IP_PROXY_HEADER* is used as client IP address.

Depending on your setup, this header might consist of several IP addresses, (for example `X-Forwarded-For: a, b, client-ip`) and you can configure which address from the header is used as client IP address here.

Advertencia: Setting this affects the security of your installation. You should only configure it to use trusted proxies for determining the IP address.

Defaults to 0.

Ver también:

Running behind reverse proxy, *Rate limiting*, `SECURE_PROXY_SSL_HEADER`, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_HEADER*

2.16.50 LEGAL_TOS_DATE

Nuevo en la versión 4.15.

Nota: You need *Información legal* installed to make this work.

Date of last update of terms of service documents. Whenever the date changes, users are required to agree with the updated terms of service.

```
from datetime import date

LEGAL_TOS_DATE = date(2022, 2, 2)
```

2.16.51 LEGAL_URL

Nuevo en la versión 3.5.

URL where your Weblate instance shows its legal documents.

Consejo: Useful if you host your legal documents outside Weblate for embedding them inside Weblate. Please check *Información legal* for details.

Ejemplo:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Ver también:

PRIVACY_URL

2.16.52 LICENSE_EXTRA

Additional licenses to include in the license choices.

Nota: Each license definition should be tuple of its short name, a long name and an URL.

Por ejemplo:

```
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

Distinto en la versión 4.3: Setting this to blank value now disables license alert.

Filter list of licenses to show. This also disables the license alert when set to empty.

Nota: This filter uses the short license names.

Por ejemplo:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

Following disables the license alert:

```
LICENSE_FILTER = set()
```

Ver también:

alerts

2.16.54 LICENSE_REQUIRED

Defines whether the license attribute in *Configuración de componentes* is required.

Nota: This is off by default.

2.16.55 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Whether the length of a given translation should be limited. The restriction is the length of the source string × 10 characters.

Consejo: Set this to `False` to allow longer translations (up to 10,000 characters) irrespective of source string length.

Nota: Defaults to `True`.

2.16.56 LOCALIZE_CDN_URL y LOCALIZE_CDN_PATH

These settings configure the *CDN de regionalización de JavaScript* add-on. *LOCALIZE_CDN_URL* defines root URL where the localization CDN is available and *LOCALIZE_CDN_PATH* defines path where Weblate should store generated files which will be served at the *LOCALIZE_CDN_URL*.

Consejo: On Hosted Weblate, this uses `https://weblate-cdn.com/`.

Ver también:

CDN de regionalización de JavaScript

2.16.57 LOGIN_REQUIRED_URLS

A list of URLs you want to require signing in. (Besides the standard rules built into Weblate).

Consejo: Esto le permite proteger con contraseña una instalación completa a través de:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Consejo: It is desirable to lock down API access as well, as shown in the above example.

Ver también:

REQUIRE_LOGIN

2.16.58 LOGIN_REQUIRED_URLS_EXCEPTIONS

List of exceptions for *LOGIN_REQUIRED_URLS*. If not specified, users are allowed to access the sign-in page.

Algunas de las excepciones que quizá quiera incluir:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$",  # Required for sign-in
    r"/static/(.*)$",  # Required for development mode
    r"/widgets/(.*)$",  # Allowing public access to widgets
    r"/data/(.*)$",  # Allowing public access to data exports
    r"/hooks/(.*)$",  # Allowing public access to notification hooks
    r"/api/(.*)$",  # Allowing access to API
    r"/js/i18n/$",  # JavaScript localization
)
```

2.16.59 MATOMO_SITE_ID

ID of the site in Matomo (formerly Piwik) you want to use for tracking Weblate.

Nota: Esta integración no admite el gestor de etiquetas de Matomo.

Ver también:

MATOMO_URL

2.16.60 MATOMO_URL

URL completo (incluida la barra al final) de una instalación de Matomo (antes Piwik) que quiera utilizar para hacer seguimiento del uso de Weblate. Visite <https://matomo.org/> para obtener más información.

Consejo: Esta integración no admite el gestor de etiquetas de Matomo.

Por ejemplo:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Ver también:

MATOMO_SITE_ID

2.16.61 NEARBY_MESSAGES

How many strings to show around the currently translated string. This is just a default value, users can adjust this in *Perfil de usuario*.

2.16.62 DEFAULT_PAGE_LIMIT

Nuevo en la versión 4.7.

Default number of elements to display when pagination is active.

2.16.63 PAGURE_CREDENTIALS

Nuevo en la versión 4.3.2.

List for credentials for Pagure servers.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
}
```

The configuration dictionary consists of credentials defined for each API host. The API host might be different from what you use in the web browser, for example GitHub API is accessed as `api.github.com`.

The following configuration is available for each host:

username
: API user, required.

token
: API token for the API user, required.

scheme
: Nuevo en la versión 4.18.

  Scheme override. Weblate attempts to parse scheme from the repository URL and falls backs to `https`. If you are running the API server internally, you might want to use `http` instead, but consider security.

Consejo: In the Docker container, the credentials are configured in three variables and the credentials are built out of that. An example configuration for GitHub might look like:

```
WEBLATE_GITHUB_USERNAME=api-user
WEBLATE_GITHUB_TOKEN=api-token
WEBLATE_GITHUB_HOST=api.github.com
```

Will be used as:

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "api-user",
        "token": "api-token",
    }
}
```

Ver también:

Solicitudes de fusión de Pagure, Pagure API

2.16.64 PRIVACY_URL

Nuevo en la versión 4.8.1.

La url donde su instancia de Weblate muestra su política de privacidad.

Consejo: Useful if you host your legal documents outside Weblate for embedding them inside Weblate, please check *Información legal* for details.

Ejemplo:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Ver también:

LEGAL_URL

2.16.65 PRIVATE_COMMIT_EMAIL_OPT_IN

Nuevo en la versión 4.15.

Configures whether the private commit e-mail is opt-in or opt-out (by default it is opt-in).

Ver también:

Perfil, *PRIVATE_COMMIT_EMAIL_TEMPLATE*

2.16.66 PRIVATE_COMMIT_EMAIL_TEMPLATE

Nuevo en la versión 4.15.

Template to generate private commit e-mail for an user. Defaults to `"{username}@users.noreply.{site_domain}"`.

Set to blank string to disable.

Nota: Using different commit e-mail is opt-in for users unless configured by *PRIVATE_COMMIT_EMAIL_OPT_IN*. Users can configure commit e-mail in the *Perfil*.

2.16.67 PROJECT_BACKUP_KEEP_COUNT

Nuevo en la versión 4.14.

Defines how many backups per project are kept on the server. Defaults to 3.

Ver también:

Copias de seguridad a nivel del proyecto

2.16.68 PROJECT_BACKUP_KEEP_DAYS

Nuevo en la versión 4.14.

Defines how long the project backups will be kept on the server. Defaults to 30 days.

Ver también:

Copias de seguridad a nivel del proyecto

2.16.69 PROJECT_NAME_RESTRICT_RE

Nuevo en la versión 4.15.

Defines a regular expression to restrict project naming. Any matching names will be rejected.

Ver también:

Nombre del proyecto

2.16.70 PROJECT_WEB_RESTRICT_HOST

Nuevo en la versión 4.16.2.

Reject using certain hosts in project website. Any subdomain is matched, so including `example.com` will block `test.example.com` as well. The list should contain lower case strings only, the parsed domain is lower cased before matching.

Default configuration:

```
PROJECT_WEB_RESTRICT_HOST = {"localhost"}
```

Ver también:

Sitio web del proyecto *PROJECT_WEB_RESTRICT_NUMERIC*, *PROJECT_WEB_RESTRICT_RE*,

2.16.71 PROJECT_WEB_RESTRICT_NUMERIC

Nuevo en la versión 4.16.2.

Reject using numeric IP address in project website. On by default.

Ver también:

Sitio web del proyecto *PROJECT_WEB_RESTRICT_HOST*, *PROJECT_WEB_RESTRICT_RE*,

2.16.72 PROJECT_WEB_RESTRICT_RE

Nuevo en la versión 4.15.

Defines a regular expression to restrict project websites. Any matching URLs will be rejected.

Ver también:

Sitio web del proyecto `PROJECT_WEB_RESTRICT_HOST`, `PROJECT_WEB_RESTRICT_NUMERIC`

2.16.73 RATELIMIT_ATTEMPTS

Nuevo en la versión 3.2.

Maximum number of authentication attempts before rate limiting is applied.

Defaults to 5.

Ver también:

Rate limiting, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

2.16.74 RATELIMIT_WINDOW

Nuevo en la versión 3.2.

How long authentication is accepted after rate limiting applies.

An amount of seconds, defaulting to 300 (5 minutes).

Ver también:

Rate limiting, `RATELIMIT_ATTEMPTS`, `RATELIMIT_LOCKOUT`

2.16.75 RATELIMIT_LOCKOUT

Nuevo en la versión 3.2.

How long authentication is locked after rate limiting applies.

An amount of seconds defaulting to 600 (10 minutes).

Ver también:

Rate limiting, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`

2.16.76 REGISTRATION_ALLOW_BACKENDS

Nuevo en la versión 4.1.

List of authentication backends to allow registration from. This only limits new registrations, users can still authenticate and add authentication using all configured authentication backends.

It is recommended to keep `REGISTRATION_OPEN` on while limiting registration backends, otherwise users will be able to register, but Weblate will not show links to register in the user interface.

Ejemplo:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Consejo: The backend names match names used in the URL for authentication.

Ver también:

REGISTRATION_OPEN, *Autenticación*

2.16.77 REGISTRATION_CAPTCHA

Whether registration of new accounts is protected by a CAPTCHA. Defaults to enabled.

If turned on, a CAPTCHA is added to all pages where a users enters their e-mail address:

- Registro de cuentas nuevas.
- Recuperación de contraseñas.
- Adding e-mail to an account.
- Formulario de contacto para usuarios que no han accedido a su cuenta.

2.16.78 REGISTRATION_EMAIL_MATCH

Le permite filtrar cuáles direcciones de correo pueden utilizarse para registrar una cuenta.

Defaults to `.*`, which allows any e-mail address to be registered.

Puede utilizarlo para restringir el registro de un dominio de correo electrónico:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.79 REGISTRATION_OPEN

Whether registration of new accounts is currently permitted. Defaults to enabled.

This setting affects built-in authentication by e-mail address or through the Python Social Auth (you can whitelist certain back-ends using *REGISTRATION_ALLOW_BACKENDS*).

Nota: Si se utilizan métodos de autenticación de terceros tales como *Autenticación LDAP*, tan solo oculta el formulario de registro, pero usuarios nuevos podrán aún acceder y crear cuentas.

Ver también:

REGISTRATION_ALLOW_BACKENDS, *REGISTRATION_EMAIL_MATCH*, *Autenticación*

2.16.80 REGISTRATION_REBIND

Nuevo en la versión 4.16.

Allow rebinding authentication backends for existing users. Turn this on when migrating between authentication providers.

Nota: Off by default to not allow adding other authentication backends to an existing account. Rebinding can lead to account compromise when using more third-party authentication backends.

2.16.81 REPOSITORY_ALERT_THRESHOLD

Nuevo en la versión 4.0.2.

Threshold for triggering an alert for outdated repositories, or ones that contain too many changes. Defaults to 25.

Ver también:

alerts

2.16.82 REQUIRE_LOGIN

Nuevo en la versión 4.1.

This enables *LOGIN_REQUIRED_URLS* and configures REST framework to require authentication for all API endpoints.

Nota: This is implemented in the *Configuración de muestra*. For Docker, use *WEBLATE_REQUIRE_LOGIN*.

2.16.83 SENTRY_DSN

Nuevo en la versión 3.9.

Sentry DSN to use for *Collecting error reports*.

Ver también:

Django integration for Sentry

2.16.84 SESSION_COOKIE_AGE_AUTHENTICATED

Nuevo en la versión 4.3.

Set session expiry for authenticated users. This complements `SESSION_COOKIE_AGE` which is used for unauthenticated users.

Ver también:

`SESSION_COOKIE_AGE`

2.16.85 SIMPLIFY_LANGUAGES

Use simple language codes for default language/country combinations. For example an `fr_FR` translation will use the `fr` language code. This is usually the desired behavior, as it simplifies listing languages for these default combinations.

Turn this off if you want to different translations for each variant.

2.16.86 SITE_DOMAIN

Configures site domain. Necessary to produce correct absolute links in many scopes (for example activation e-mails, notifications or RSS feeds).

If Weblate is running on a non-standard port, include it here as well.

Ejemplos:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Nota: This setting should only contain the domain name. For configuring protocol, (turning on and enforcing HTTPS) use *ENABLE_HTTPS* and for changing the URL, use *URL_PREFIX*.

Consejo: En un contenedor Docker, el dominio del sitio se configura mediante *WEBLATE_ALLOWED_HOSTS*.

Ver también:

Set correct site domain, *Allowed hosts setup*, *Correctly configure HTTPS* *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

2.16.87 SITE_TITLE

El título que se utilizará en el sitio web y los mensajes de correo que se envíen.

2.16.88 SPECIAL_CHARS

Caracteres adicionales que incluir en el teclado visual; *Visual keyboard*.

El valor predeterminado es:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "…")
```

2.16.89 SINGLE_PROJECT

Nuevo en la versión 3.8.

Redirects users directly to a project or component instead of showing the dashboard. You can either set it to `True` so it only works if there is actually only single project in Weblate. Alternatively, set the project slug, and it will redirect unconditionally to this project.

Distinto en la versión 3.11: The setting now also accepts a project slug, to force displaying that single project.

Ejemplo:

```
SINGLE_PROJECT = "test"
```

2.16.90 SSH_EXTRA_ARGS

Nuevo en la versión 4.9.

Allows adding custom parameters when Weblate is invoking SSH. Useful when connecting to servers using legacy encryption or other non-standard features.

For example when SSH connection in Weblate fails with *Unable to negotiate with legacyhost: no matching key exchange method found. Their offer: diffie-hellman-group1-sha1*, you can turn that on using:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"
```

Consejo: The string is evaluated by the shell, so ensure any whitespace and special characters is quoted.

Ver también:

OpenSSH Legacy Options

2.16.91 STATUS_URL

The URL where your Weblate instance reports its status.

2.16.92 SUGGESTION_CLEANUP_DAYS

Nuevo en la versión 3.2.1.

Automatically deletes suggestions after a given number of days. Defaults to `None`, meaning no deletions.

2.16.93 UNUSED_ALERT_DAYS

Nuevo en la versión 4.17.

Configures when the *Component seems unused* alert is triggered.

Defaults to 365 days, set to 0 to turn it off.

2.16.94 UPDATE_LANGUAGES

Nuevo en la versión 4.3.2.

Controls whether languages database should be updated when running database migration and is on by default. This setting has no effect on invocation of *`weblate weblate setuplang`*.

Advertencia: The languages display might become inconsistent with this. Weblate language definitions expand over time and it will not display language code for the defined languages.

Ver también:

Definiciones de idioma incorporadas

2.16.95 URL_PREFIX

This setting allows you to run Weblate under some path (otherwise it relies on being run from the webserver root).

Nota: To use this setting, you also need to configure your server to strip this prefix. For example with WSGI, this can be achieved by setting `WSGIScriptAlias`.

Consejo: The prefix should start with a `/`.

Ejemplo:

```
URL_PREFIX = "/translations"
```

Nota: This setting does not work with Django's built-in server, you would have to adjust `urls.py` to contain this prefix.

2.16.96 VCS_API_DELAY

Nuevo en la versión 4.15.1.

Configures minimal delay in seconds between third-party API calls in *Solicitudes de incorporación de GitHub*, *Solicitudes de fusión de GitLab*, *Solicitudes de incorporación de Gitea*, and *Solicitudes de fusión de Pagure*.

This rate-limits API calls from Weblate to these services to avoid overloading them.

If you are being limited by secondary rate-limiter at GitHub, increasing this might help.

The default value is 10.

2.16.97 VCS_BACKENDS

Configuration of available VCS backends.

Nota: Weblate tries to use all supported back-ends you have the tools for.

Consejo: You can limit choices or add custom VCS back-ends by using this.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Ver también:

Integración de control de versiones

2.16.98 VCS_CLONE_DEPTH

Nuevo en la versión 3.10.2.

Configures how deep cloning of repositories Weblate should do.

Nota: Currently this is only supported in *Git*. By default Weblate does shallow clones of the repositories to make cloning faster and save disk space. Depending on your usage (for example when using custom *Complementos*), you might want to increase the depth or turn off shallow clones completely by setting this to 0.

Consejo: In case you get `fatal: protocol error: expected old/new/ref, got 'shallow <commit hash>'` error when pushing from Weblate, turn off shallow clones completely by setting:

```
VCS_CLONE_DEPTH = 0
```

2.16.99 WEBLATE_ADDONS

List of add-ons available for use. To use them, they have to be enabled for a given translation component. By default this includes all built-in add-ons, when extending the list you will probably want to keep existing ones enabled, for example:

```
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",
)
```

Nota: Removing the add-on from the list does not uninstall it from the components. Weblate will crash in that case. Please uninstall the add-on from all components prior to removing it from this list.

Ver también:

Complementos, *DEFAULT_ADDONS*

2.16.100 WEBLATE_EXPORTERS

Nuevo en la versión 4.2.

List of a available exporters offering downloading translations or glossaries in various file formats.

Ver también:

Formatos de archivo admitidos

2.16.101 WEBLATE_FORMATS

Nuevo en la versión 3.0.

List of file formats available for use.

Nota: The default list already has the common formats.

Ver también:

Formatos de archivo admitidos

2.16.102 WEBLATE_MACHINERY

Nuevo en la versión 4.13.

List of machinery services available for use.

Ver también:

Configuración de las sugerencias automáticas

2.16.103 WEBLATE_GPG_IDENTITY

Nuevo en la versión 3.1.

Identity used by Weblate to sign Git commits, for example:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

The Weblate GPG keyring is searched for a matching key (`home/.gnupg` under *DATA_DIR*). If not found, a key is generated, please check *Signing Git commits with GnuPG* for more details.

Ver también:

Signing Git commits with GnuPG

2.16.104 WEBSITE_REQUIRED

Defines whether *Sitio web del proyecto* has to be specified when creating a project. On by default, as that suits public server setups.

2.17 Configuración de muestra

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"

# 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",
```

(continué en la próxima página)

(proviene de la página anterior)

```
            # 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
# 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", "ةيبرعلا"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be-latn", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("cy", "Cymraeg"),
    ("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", "□□□"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "□□□"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
```

(continué en la próxima página)

(proviene de la página anterior)

```
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "□□□"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "□□□□"),
    ("zh-hant", "□□□□"),
)

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

# 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(CACHE_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",
)
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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",
            ],
        },
        "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.github_enterprise.GithubEnterpriseOAuth2",
    # "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 = ""
```

(continué en la próxima página)

(proviene de la página anterior)

```
SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_GITHUB_ENTERPRISE_KEY = ""
SOCIAL_AUTH_GITHUB_ENTERPRISE_SECRET = ""
SOCIAL_AUTH_GITHUB_ENTERPRISE_URL = ""
SOCIAL_AUTH_GITHUB_ENTERPRISE_API_URL = ""
SOCIAL_AUTH_GITHUB_ENTERPRISE_SCOPE = ""

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 = (
    "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
```

(continué en la próxima página)

(proviene de la página anterior)

```
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"
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {"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",
```

(continué en la próxima página)

(proviene de la página anterior)

```
    "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",
    "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"
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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)
        # Since Python 3.7 connect failures are silently discarded, so
        # the exception is almost never raised here. Instead we look whether the␣
↪socket
        # to syslog is open after init.
        HAVE_SYSLOG = handler.socket.fileno() != -1
        handler.close()
    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
# 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",
```

(continué en la próxima página)

(proviene de la página anterior)

```
        },
        "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,
        },
        # Logging database queries
        # "django.db.backends": {
        #     "handlers": [DEFAULT_LOG],
        #     "level": "DEBUG",
        # },
        "redis_lock": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        "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",
```

(continué en la próxima página)

(proviene de la página anterior)

```
    #     "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

# 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
```

(continué en la próxima página)

(proviene de la página anterior)

```
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
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
```

(continué en la próxima página)

(proviene de la página anterior)

```
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
# 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",
```

(continué en la próxima página)

(proviene de la página anterior)

```
#     "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.ReusedCheck",
#     "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",
#     "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",
#     "weblate.trans.autofixes.chars.DevanagariDanda",
#     "weblate.trans.autofixes.html.BleachHTML",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
```

(continué en la próxima página)

(proviene de la página anterior)

```
#     "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.generate.FillReadOnlyAddon",
#     "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"

# 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(CACHE_DIR, "avatar"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
```

(continué en la próxima página)

(proviene de la página anterior)

```
}

# 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
FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = "weblate.utils.compress.offline_context"
COMPRESS_CSS_HASHING_METHOD = "content"

# 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
# )
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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"},
}

# 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 Órdenes de gestión

Nota: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check *Permisos del sistema de archivos* 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 Invocar órdenes de gestión

Como ya se mencionó, la invocación dependerá de cómo instaló Weblate.

If using virtualenv for Weblate, you can either specify the full path to **`weblate`**, or activate the virtualenv prior to invoking it:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

If you are using source code directly (either from a tarball or Git checkout), the management script is `./manage.py` available in the Weblate sources. To run it:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip installer, or by using the `./setup.py` script, the **`weblate`** is installed to your path (or virtualenv path), from where you can use it to control Weblate:

```
weblate list_versions
```

For the Docker image, the script is installed like above, and you can run it using **`docker exec`**:

```
docker exec --user weblate <container> weblate list_versions
```

For **`docker-compose-plugin`** 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
```

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
```

Ver también:

Instalar con Docker, *Instalar en Debian y Ubuntu*, *Instalar en SUSE y openSUSE*, *Instalar en Red Hat, Fedora y CentOS*, *Instalar desde el código fuente*

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

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).

Ejemplo:

```
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>

Distinto en la versión 4.6: Parámetro agregado para el modo de traducción.

Realiza traducciones automáticas basadas en las traducciones de otros componentes.

--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 *Incoherente*).

--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.

Ejemplo:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application␣
↪weblate website cs
```

Ver también:

Traducción automática

2.18.4 celery_queues

weblate celery_queues

Nuevo en la versión 3.7.

Displays length of Celery task queues.

2.18.5 checkgit

weblate checkgit <project|project/component>

Prints current state of the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

2.18.6 commitgit

weblate commitgit <project|project/component>

Commits any possible pending changes to the back-end Git repository.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components, or use `--file-format` to filter based on the file format.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Commits pending changes older than a given age.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

--age HOURS

Age in hours for committing. If not specified the value configured in *Configuración de componentes* is used.

Nota: 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 *Configuración de componentes*.

Ver también:

Efectuar tareas de mantenimiento, *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.

Ver también:

Efectuar tareas de mantenimiento

2.18.9 cleanup_ssh_keys

weblate cleanup_ssh_keys

Nuevo en la versión 4.9.1.

Performs cleanup of stored SSH host keys:

- Removes deprecated RSA keys for GitHub which might cause issues connecting to GitHub.
- Removes duplicate entries in host keys.

Ver también:

Repositorios SSH

2.18.10 createadmin

weblate createadmin

Creates an `admin` account with a random password, unless it is specified.

--password PASSWORD
: Permite proporcionar una contraseña mediante la línea de órdenes, de modo que no se genere una aleatoria.

--no-password
: Do not set password, this can be useful with *–update*.

--username USERNAME
: Use the given name instead of `admin`.

--email USER@EXAMPLE.COM
: Specify the admin e-mail address.

--name
: Specify the admin name (visible).

--update
: Update the existing user (you can use this to change passwords).

2.18.11 dump_memory

weblate dump_memory

Export a JSON file containing Weblate Translation Memory content.

Ver también:

Memoria de traducción, *Weblate Translation Memory Schema*

2.18.12 dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by *weblate weblate importuserdata*.

Consejo: Esto resulta útil cuando hay que migrar o fusionar instalaciones de Weblate.

2.18.13 import_demo

weblate import_demo

Nuevo en la versión 4.1.

Creates a demo project with components based on <https://github.com/WeblateOrg/demo>. Make sure the celery tasks are running before running this command.

Puede ser de utilidad si está desarrollando Weblate.

2.18.14 import_json

weblate import_json <json-file>

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.

Ejemplo de archivo JSON:

```
[
  {
    "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"
  }
]
```

Ver también:

weblate weblate import_memory

2.18.15 import_memory

weblate import_memory <file>

Importa un archivo TMX o JSON en la memoria de traducción de Weblate.

--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.

Ver también:

Memoria de traducción, *Weblate Translation Memory Schema*

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

Distinto en la versión 3.0: The import_project command is now based on the *Detección de componentes* 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 *Formatos de archivo admitidos*), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see *Configuración de componentes*) 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'
```

Ver también:

More detailed examples can be found in the starting chapter, alternatively you might want to use *weblate weblate import_json*.

2.18.17 importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by *weblate weblate 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

Nuevo en la versión 3.2.

weblate install_addon --addon ADDON <project|project/component>

Instala un complemento a un conjunto de componentes.

--addon ADDON

Name of the add-on to install. For example `weblate.gettext.customize`.

--configuration CONFIG

Configuración codificada en JSON de un complemento.

--update

Actualice la configuración del complemento existente.

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 *Personalizar la salida de gettext* for all components:

```
weblate install_addon --addon weblate.gettext.customize --configuration '{"width":␣
↪-1}' --update --all
```

Ver también:

Complementos

2.18.20 install_machinery

Nuevo en la versión 4.18.

weblate install_machinery --service SERVICE

Installs an site-wide automatic suggestion service.

--service SERVICE
: Name of the service to install. For example `deepl`.

--configuration CONFIG
: JSON encoded configuration of a service.

--update
: Update the existing service configuration.

To install *DeepL*:

```
weblate install_service --service deepl --configuration '{"key": "x", "url":
↪"https://api.deepl.com/v2/"}' --update
```

Ver también:

Configuración de las sugerencias automáticas

2.18.21 list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names.

This is used to generate <https://wiki.l10n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>.

2.18.22 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.

2.18.23 list_versions

weblate list_versions

Lists all Weblate dependencies and their versions.

2.18.24 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.

Nota: 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.

2.18.25 lock_translation

weblate lock_translation <project|project/component>

Prevents further translation of a component.

Consejo: 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.

Ver también:

weblate weblate unlock_translation

2.18.26 migrate

weblate migrate

Migrates database to current Weblate schema. The command line options are described at Django `migrate`.

Consejo: In case you want to run an installation non interactively, you can use `weblate migrate --noinput`, and then create an admin user using *weblate weblate createadmin* command.

Ver también:

`migrate`, *Filling up the database*

2.18.27 move_language

weblate move_language source target

Nuevo en la versión 3.0.

Allows you to merge language content. This is useful when updating to a new version which contains aliases for previously unknown languages that have been created with the *(generated)* suffix. It moves all content from the *source* language to the *target* one.

Ejemplo:

```
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.

2.18.28 pushgit

weblate pushgit <project|project/component>

Pushes committed changes to the upstream VCS repository.

--force-commit

Force commits any pending changes, prior to pushing.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Nota: Weblate pushes changes automatically if *Enviar al consignar* in *Configuración de componentes* is turned on, which is the default.

2.18.29 unlock_translation

weblate unlock_translation <project|project/component>

Desbloquea el componente que se indique para que se pueda traducir.

Consejo: 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.

Ver también:

weblate weblate lock_translation

2.18.30 setupgroups

weblate setupgroups

Configura los grupos predeterminados y, opcionalmente, asigna todos los usuarios a ese grupo predeterminado.

--no-privs-update

Desactiva la actualización automática de los grupos existentes (solamente añade los nuevos).

--no-projects-update

Prevents automatic updates of groups for existing projects. This allows adding newly added groups to existing projects, see *Control de acceso al proyecto*.

Ver también:

Lista de privilegios y roles incorporados

2.18.31 setuplang

weblate setuplang

Actualiza la lista de idiomas definidos en Weblate.

--no-update

Desactiva la actualización automática de los idiomas existentes (solamente añade los nuevos).

2.18.32 updatechecks

weblate updatechecks <project|project/component>

Actualiza todas las comprobaciones para todas las cadenas.

Consejo: 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.

Nota: Checks are recalculated regularly by Weblate in the background, the frequency can be configured via *BACKGROUND_TASKS*.

2.18.33 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.

Nota: Usually it is better to configure hooks in the repository to trigger *Actuadores de notificación*, instead of regular triggering the updates by *weblate weblate updategit*.

2.19 Anuncios

Distinto en la versión 4.0: En versiones anteriores, esta prestación se denominaba «mensajes en pizarra».

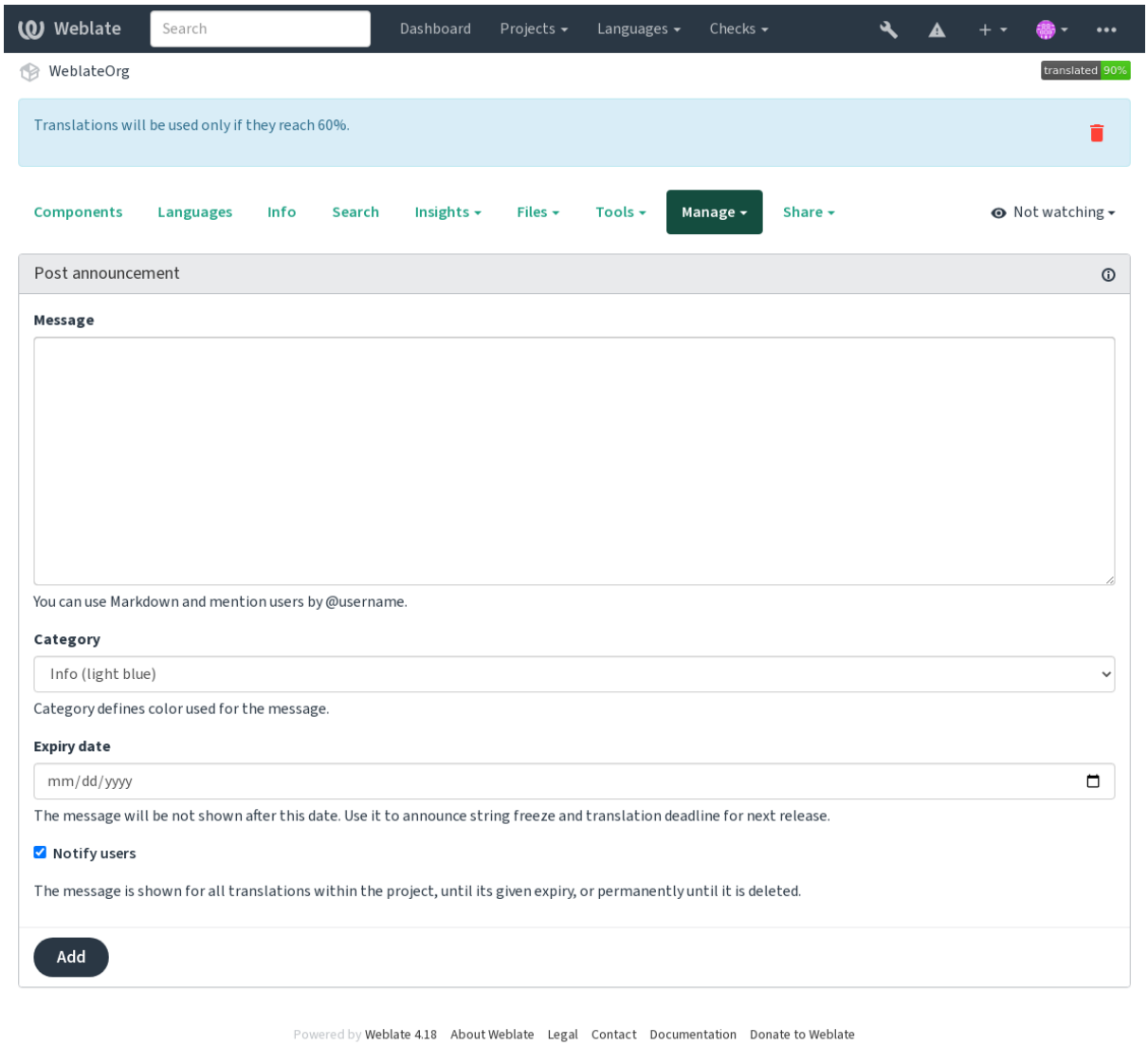
Proporciona información a tus traductores publicando anuncios, en todo el sitio, por proyecto, componente o idioma.

Anunciar el propósito, los plazos, el estado, o especificar los objetivos de la traducción.

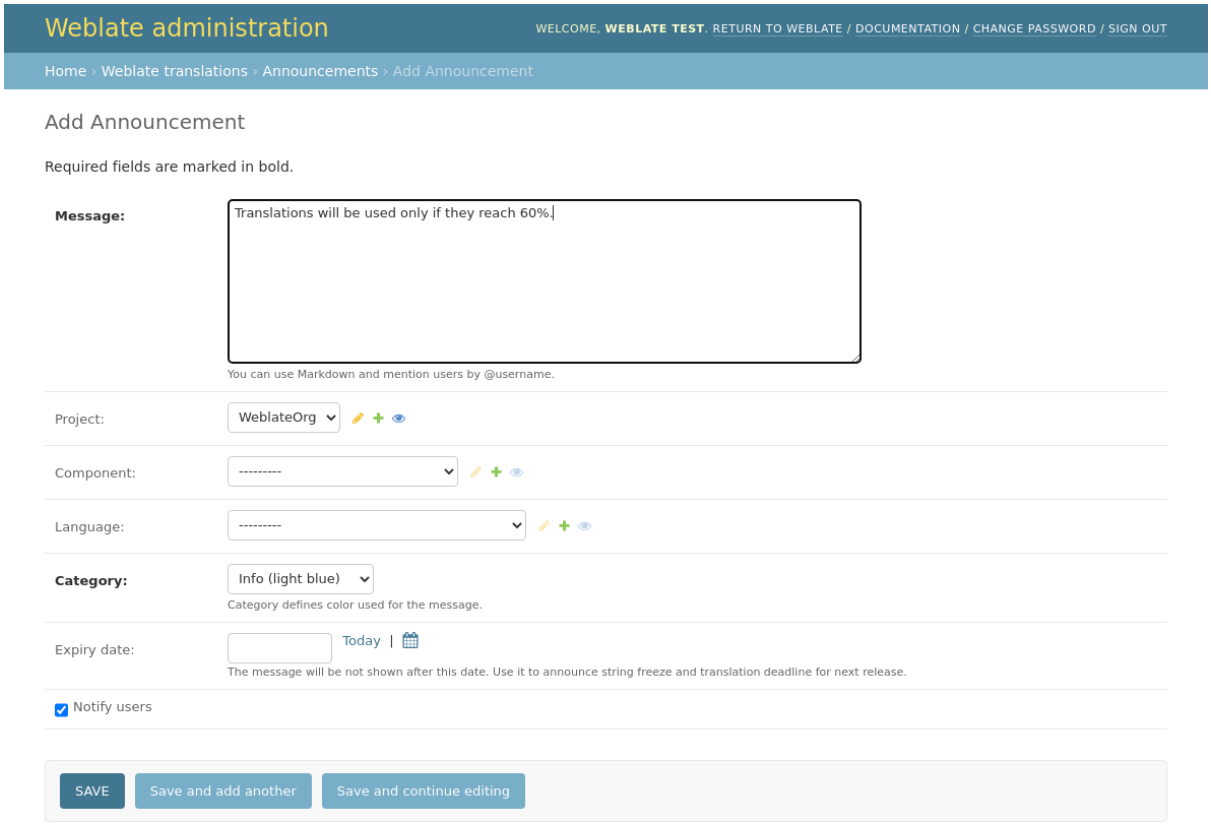
Los usuarios recibirán una notificación por cada anuncio en los proyectos que monitorizan (salvo que decidan no recibirlas).

Esto puede ser útil para varias cosas, desde anunciar el propósito del sitio web hasta especificar los objetivos de las traducciones.

Se pueden publicar los anuncios en cada nivel en el menú *Manage*, utilizando *Post announcement*:



También se puede añadir mediante la interfaz de administración:



A continuación, los anuncios se muestran en función de su contexto especificado:

Ningún contexto especificado

Se muestra en el panel de herramientas (página de destino).

Proyecto especificado

Se muestra dentro del proyecto, incluyendo todos sus componentes y traducciones.

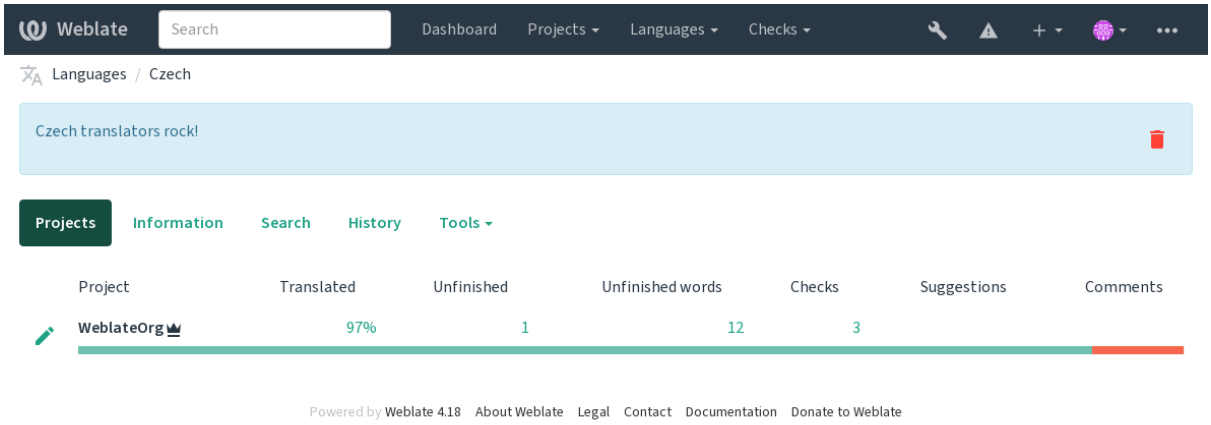
Componente especificado

Se muestra para un componente determinado y todas sus traducciones.

Idioma especificado

Se muestra en el resumen de idiomas y en todas las traducciones en ese idioma.

Así es como se ve en la página de resumen de idiomas:



2.20 Listas de componentes

Especifica varias listas de componentes para que aparezcan como opciones en el panel de control del usuario, de las cuales los usuarios pueden elegir una como vista predeterminada. Para saber más, ver *Cuadro de mando*.

Consejo: Se presentará un estado para cada lista de componentes presentada en el tablero.

Los nombres y el contenido de las listas de componentes pueden especificarse en la interfaz de administración, en la sección: *Component lists*. Cada lista de componentes debe tener un nombre que se muestre al usuario, y un slug que lo represente en la URL.

Consejo: Cambia la configuración del panel de control para los usuarios anónimos desde la interfaz de administración, modificando el panel de control que se presenta a los usuarios no autentificados.

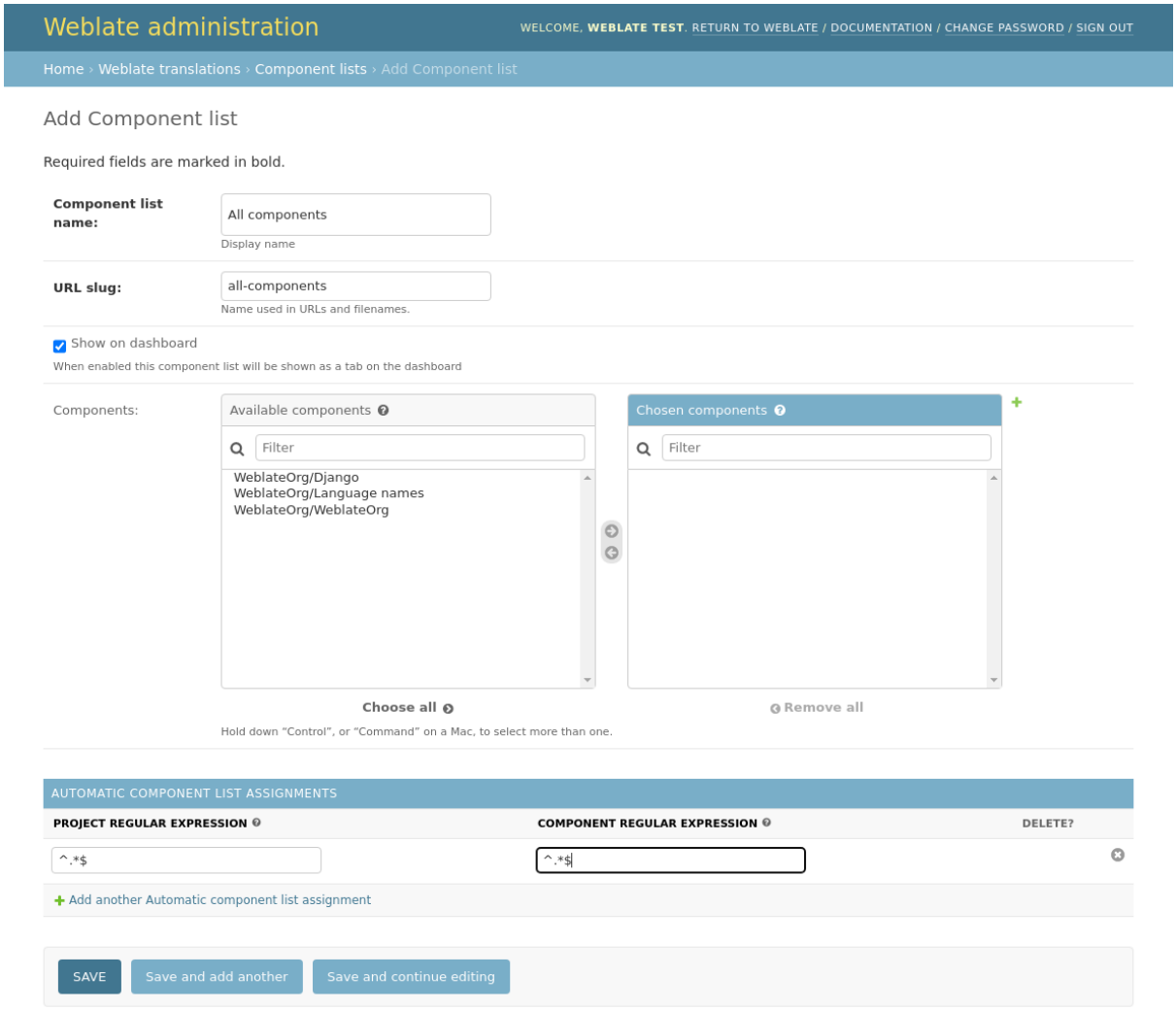
2.20.1 Listas de componentes automáticas

Añadir componentes a la lista automáticamente en función de su slug creando reglas: *Automatic component list assignment*.

- Es útil para mantener listas de componentes para instalaciones grandes, o en caso de que quieras tener una lista de componentes con todos los componentes de tu instalación de Weblate.

Consejo: Haz una lista de componentes que contenga todos los componentes de tu instalación de Weblate.

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:



2.21 Módulos opcionales de Weblate

Hay varios módulos opcionales disponibles para que los configure.

2.21.1 Git exporter

Provides you read-only access to the underlying Git repository using HTTP(S).

Instalación

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
```

Uso

The module automatically hooks into Weblate and sets the exported repository URL in the *Configuración de componentes*. 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/'
```

Consejo: 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 Facturación

This is used on Hosted Weblate to define billing plans, track invoices and usage limits.

Instalación

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
```

Uso

After installation you can control billing in the admin interface. Users with billing enabled will get new *Billing* tab in their *Perfil de usuario*.

The billing module additionally allows project admins to create new projects and components without being superusers (see *Añadir proyectos y componentes de traducción*). 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 Información legal

This is used on Hosted Weblate to provide required legal documents. It comes provided with blank documents, and you are expected to fill out the following templates in the documents:

`legal/documents/tos.html`
Terms of service document

`legal/documents/privacy.html`
Privacy policy document

`legal/documents/summary.html`
Short overview of the terms of service and privacy policy

On changing the terms of service documents, please adjust `LEGAL_TOS_DATE` so that users are forced to agree with the updated documents.

Nota: 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.

Instalación

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.

Uso

After installation and editing, the legal documents are shown in the Weblate UI.

2.21.4 Avatars

Avatars are downloaded and cached server-side to reduce information leaks to the sites serving them by default. The built-in support for fetching avatars from e-mails addresses configured for it can be turned off using `ENABLE_AVATARS`.

Weblate currently supports:

- Gravatar
- Libravatar

Ver también:

Caché de avatars, `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:

- Sugerencias de usuarios no autenticados
- Descripciones y enlaces de proyectos y componentes

Nota: This (among other things) relies on IP address of the client, please see *Running behind reverse proxy* for properly configuring that.

Ver también:

Running behind reverse proxy, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Signing Git commits with GnuPG

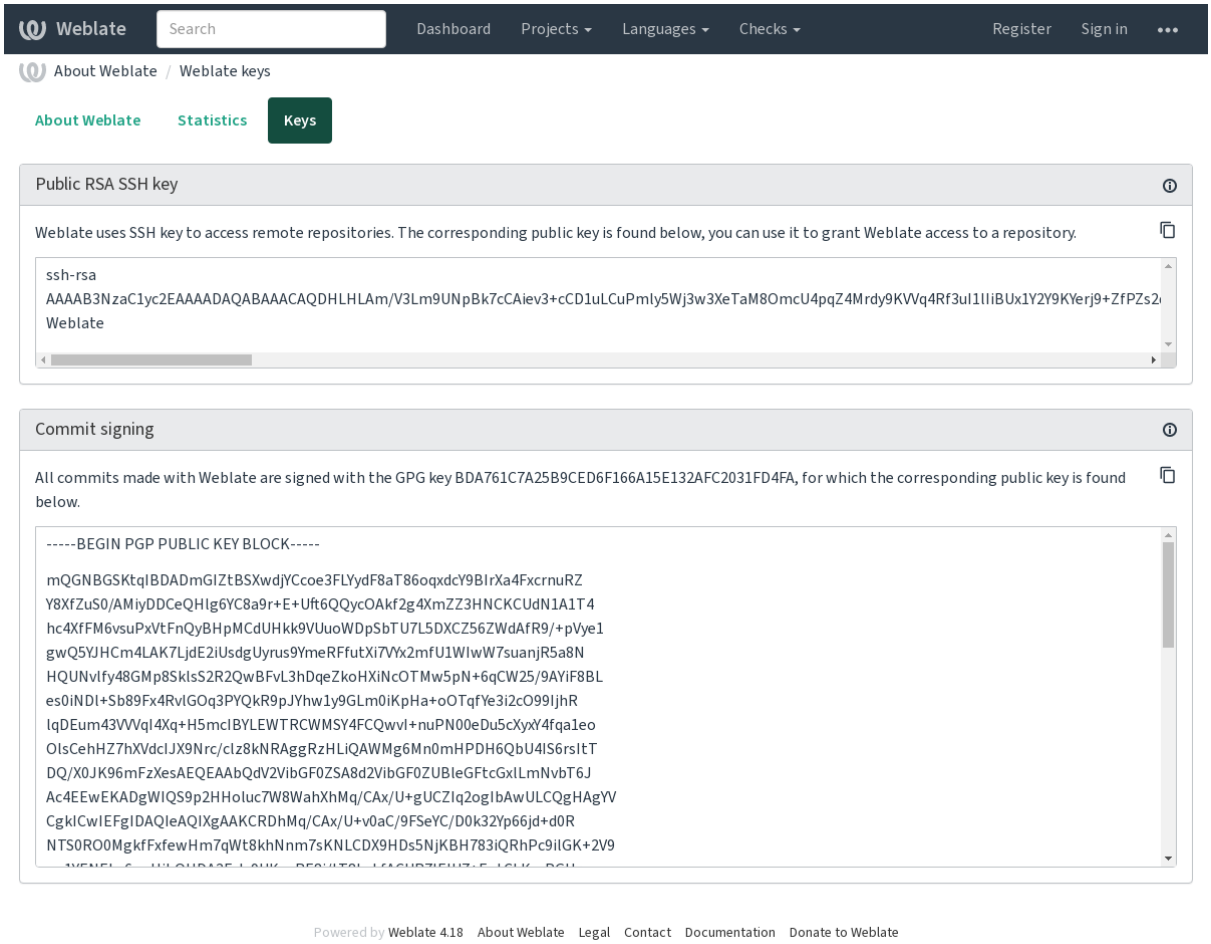
Nuevo en la versión 3.1.

All commits can be signed by the GnuPG key of the Weblate instance.

1. Turn on `WEBLATE_GPG_IDENTITY`. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

This feature needs GnuPG 2.1 or newer installed.

You can find the key in the `DATA_DIR` and the public key is shown on the «About» page:



2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking gpg.

Ver también:

WEBLATE_GPG_IDENTITY

2.21.7 Rate limiting

Distinto en la versión 3.2: The rate limiting now accepts more fine-grained configuration.

Distinto en la versión 4.6: The rate limiting no longer applies to superusers.

Several operations in Weblate are rate limited. At most *RATELIMIT_ATTEMPTS* attempts are allowed within *RATELIMIT_WINDOW* seconds. The user is then blocked for *RATELIMIT_LOCKOUT*. There are also settings specific to scopes, for example `RATELIMIT_CONTACT_ATTEMPTS` or `RATELIMIT_TRANSLATE_ATTEMPTS`. The table below is a full list of available scopes.

The following operations are subject to rate limiting:

Nombre	Alcance	Intentos permitidos	Oportunidad del límite de velocidad	Período de bloqueo
Registro	`REGISTRATION`	5	300	600
Enviar un mensaje a los administradores	`MESSAGE`	2	300	600
Autenticación de la contraseña al iniciar la sesión	`LOGIN`	5	300	600
Búsqueda en todo el sitio	`SEARCH`	6	60	60
Traducción	`TRANSLATE`	30	60	600
Adición al glosario	`GLOSSARY`	30	60	600
Comenzar una traducción en un idioma nuevo	`LANGUAGE`	2	300	600
Crear un nuevo proyecto	`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 *Limitación de la tasa de API*.

Ver también:

Rate limiting, *Running behind reverse proxy*, *Limitación de la tasa de API*

2.21.8 Integración con Fedora Messaging

Fedora Messaging es un publicador basado en AMQP para todos los cambios que ocurren en Weblate. Puede integrar servicios adicionales para actuar ante cambios que ocurran en Weblate mediante esta herramienta.

La integración con Fedora Messaging está disponible como un módulo Python separado, `weblate-fedora-messaging`. Consulte <https://github.com/WeblateOrg/fedora_messaging/> para obtener instrucciones de instalación.

2.22 Personalizar Weblate

Extend and customize using Django and Python. Contribute your changes upstream so that everybody can benefit. This reduces your maintenance costs; code in Weblate is taken care of when changing internal interfaces or refactoring the code.

Advertencia: Neither internal interfaces nor templates are considered a stable API. Please review your own customizations for every upgrade, the interfaces or their semantics might change without notice.

Ver también:

Contribuir con Weblate

2.22.1 Crear un módulo Python

If you are not familiar with Python, you might want to look into Python For Beginners, explaining the basics and pointing to further tutorials.

To write a file with custom Python code (called a module), a place to store it is needed, either in the system path (usually something like `/usr/lib/python3.9/site-packages/`) or in the Weblate directory, which is also added to the interpreter search path.

Nuevo en la versión 3.8-5: When *using Docker*, you can place Python modules in `/app/data/python/` (see *Volúmenes de contenedores Docker*), so they can be loaded by Weblate, for example from a *settings override file*.

Better yet, turn your customization into a proper Python package:

1. Create a folder for your package (we will use *weblate_customization*).
2. Within it, create a `setup.py` file to describe the package:

```
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. Create a folder for the Python module (also called `weblate_customization`) for the customization code.
4. Within it, create a `__init__.py` file to make sure Python can import the module.
5. This package can now be installed using **`pip install -e`**. More info to be found in Editable installs.
6. Once installed, the module can be used in the Weblate configuration (for example `weblate_customization.checks.FooCheck`).

Your package structure should look like this:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

You can find an example of customizing Weblate at <https://github.com/WeblateOrg/customize-example>, it covers all the topics described below.

2.22.2 Cambiar el logo

1. Create a simple Django app containing the static files you want to overwrite (see *Crear un módulo Python*).

   Branding appears in the following files:

   `icons/weblate.svg`
   : Logo mostrado en la barra de navegación.

   `logo-*.png`
   : Web icons depending on screen resolution and web-browser.

   `favicon.ico`
   : Icono web empleado por navegadores obsoletos.

weblate-*.png
Avatars for bots or anonymous users. Some web-browsers use these as shortcut icons.

email-logo.png
Used in notifications e-mails.

2. Add it to `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here…
)
```

3. Run `weblate collectstatic --noinput`, to collect static files served to clients.

Ver también:

How to manage static files (e.g. images, JavaScript, CSS), *Serving static files*

2.22.3 Controles de calidad personalizados, complementos y autocorrecciones

To install your code for *Correcciones automáticas personalizadas*, *Escribir los propios controles* or *Complemento de escritura* in Weblate:

1. Place the files into your Python module containing the Weblate customization (see *Crear un módulo Python*).
2. Add its fully-qualified path to the Python class in the dedicated settings (*WEBLATE_ADDONS*, *CHECK_LIST* or *AUTOFIX_LIST*):

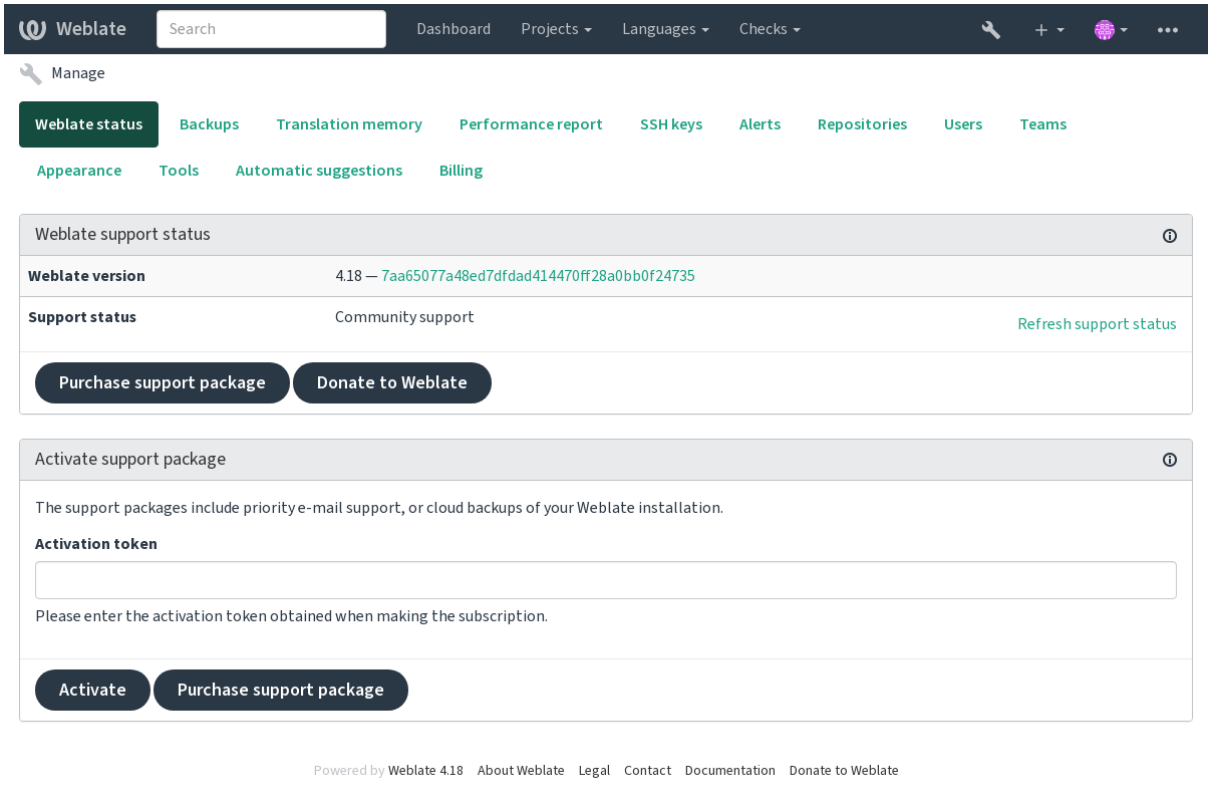
```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Ver también:

Correcciones automáticas personalizadas, *Escribir los propios controles*, *Complemento de escritura*, *Ejecutar scripts desde el complemento*

2.23 Interfaz de gestión

La interfaz de gestión ofrece ajustes de administración bajo la URL `/manage/`. Está disponible para los usuarios que hayan iniciado sesión con privilegios de administrador, accediendo a través del icono de la llave inglesa en la parte superior derecha:



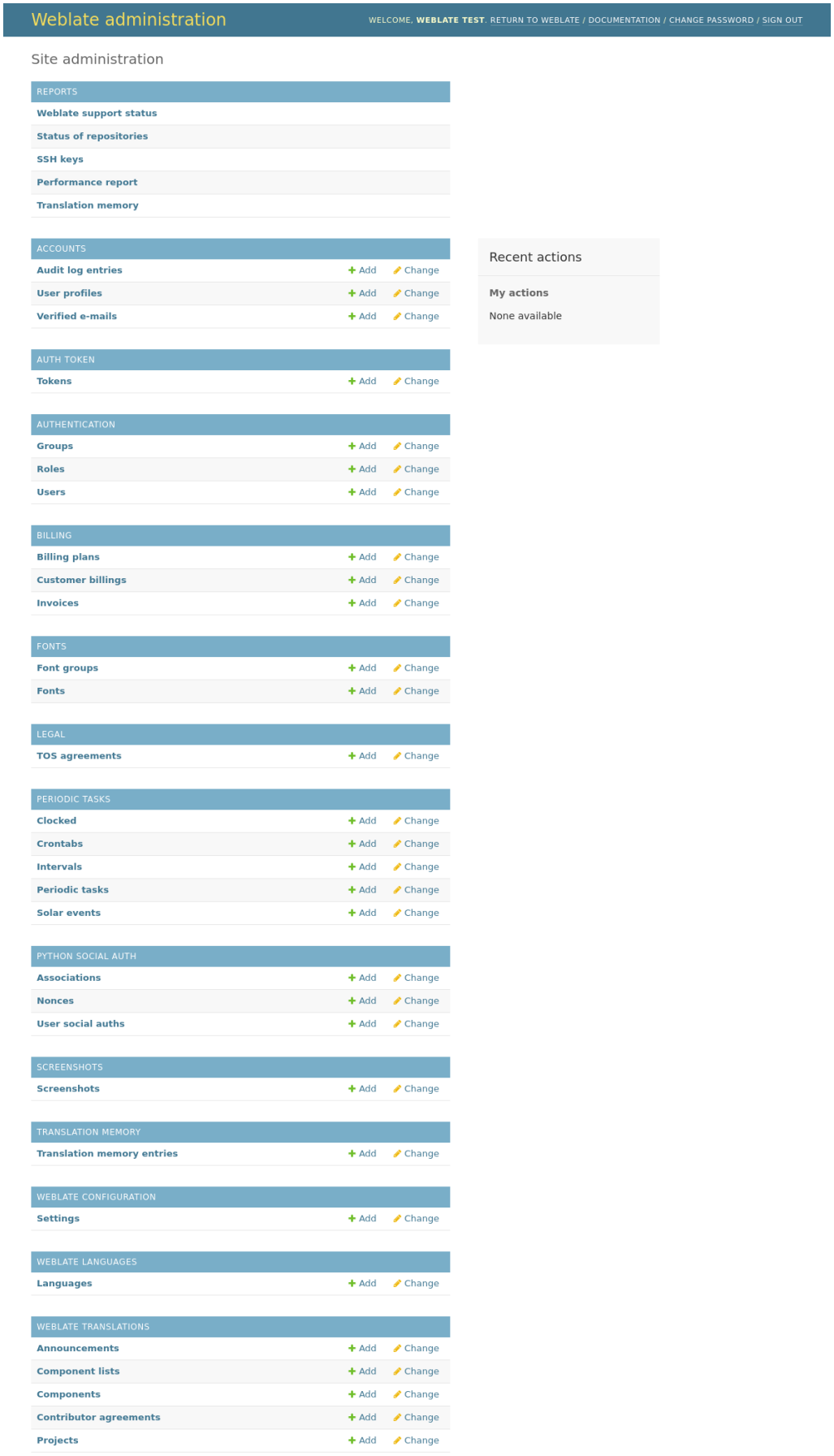
Incluye un resumen básico de tu Weblate:

- Estado de servicio técnico; vea *Obtener ayuda con Weblate*
- Copias de respaldo; vea *Respaldar y trasladar Weblate*
- Memoria de traducción compartida; vea *Memoria de traducción*
- Informe de rendimiento para revisar el funcionamiento de Weblate y la duración de las consultas de Celery
- Gestión de claves SSH; vea *Repositorios SSH*
- Resumen de alertas para todos los componentes, consulte alerts

2.23.1 La interfaz administrativa de Django

Advertencia: Use con precaución, ya que esta es una interfaz de bajo nivel. No debería necesitarlo en la mayoría de los casos, puesto que la mayoría de las cosas son accesibles cómodamente a través de la API o la interfaz de usuario de Weblate.

Aquí puede gestionar los objetos almacenados en la base de datos, tales como usuarios, traducciones y determinadas configuraciones:


Weblate administration
WELCOME, WEBLATE TEST. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Site administration
REPORTS
Weblate support status
Status of repositories
SSH keys
Performance report
Translation memory
ACCOUNTS
Audit log entries + 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

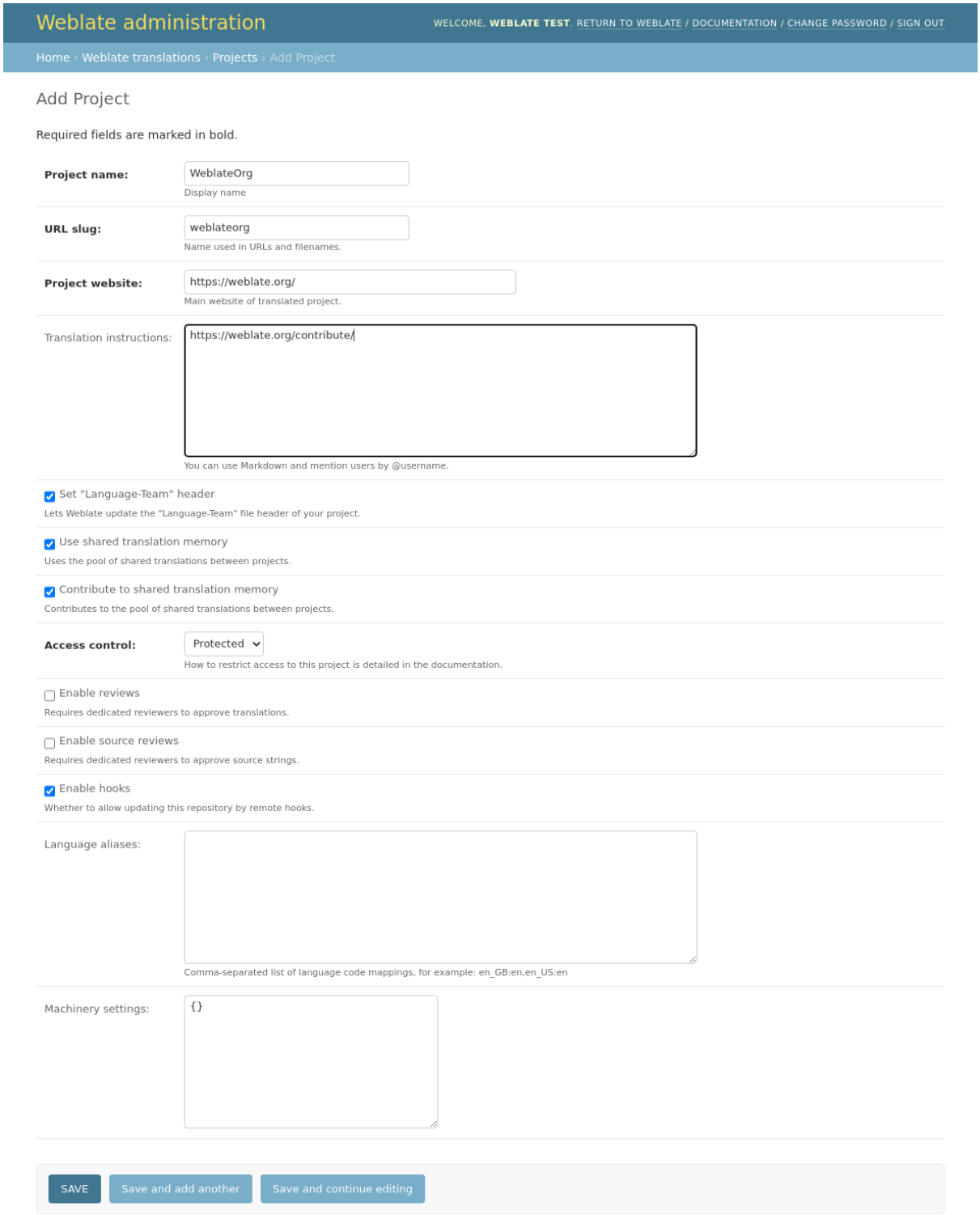
En la sección *Reports*, puedes comprobar el estado de tu sitio, ajústalo para *Puesta en marcha de entorno de producción*, o gestiona las claves SSH utilizadas para acceder a *Accessing repositories*.

Gestiona los objetos de la base de datos en cualquiera de las secciones. La más interesante es probablemente *Weblate translations*, donde puedes gestionar proyectos traducibles, ver *Configuración de proyectos* y *Configuración de componentes*.

Weblate languages contiene las definiciones de los idiomas, que se explican en *Definiciones de idioma*.

Añadir un proyecto

Añadir un proyecto sirve de contenedor para todos los componentes. Normalmente se crea un proyecto para una pieza de software, o libro (Ver *Configuración de proyectos* para información sobre los parámetros individuales):



Ver también:

Configuración de proyectos

Componentes bilingües

Una vez añadido un proyecto, se pueden añadir componentes de traducción al mismo. (Véase *Configuración de componentes* para obtener información sobre los parámetros individuales):

Weblate administration
WELCOME, WEBLATE TEST RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Home › Weblate translations › Components › Add Component
Add Component
IMPORT SPEED DOCUMENTATION
Required fields are marked in bold.
Component name:
Language names
Display name.
URL slug:
language-names
Name used in URLs and filenames.
Project:
WeblateOrg
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 push URL:
URL of a push repository, pushing is turned off if empty.
Repository browser:
https://github.com/WeblateOrg/demo/blob/{{branch}}/{{fil
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}}.
Exported repository URL:
URL of repository where users can fetch changes from Weblate
Source string bug reporting address:
E-mail address for reports on errors in source strings. Leave empty for no e-mails.
Repository branch:
Repository branch to translate
Push branch:
Branch for pushing changes, leave empty to use repository branch
File mask:
weblate/langdata/locale/%LC_MESSAGES/
Path of files to translate relative to repository root, use * instead of language code, for example: po/*.po or locale/%LC_MESSAGES/django.po.
Screenshot file mask:
Path of screenshots relative to repository root, for example: docs/screenshots/*.png.
Monolingual base language file:
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.
Template for new translations:
weblate/langdata/locale/django.pot
Filename of file used for creating new translations. For gettext choose .pot file.
File format:
gettext PO file
Locked
Locked component will not get any translation updates.
Allow translation propagation
Whether translation updates in other components will cause automatic translation in this one
Turn on suggestions
Whether to allow translation suggestions at all.
Suggestion voting
Users can only vote for suggestions and can't make direct translations.
Autoaccept suggestions:
0
Automatically accept suggestions with this number of votes, use 0 to turn it off.
Translation flags:
Additional comma-separated flags to influence Weblate behavior.
Enforced checks:
[]
List of checks which can not be ignored.
Translation license:
GNU General Public License v3.0 or later
Contributor agreement:
User agreement which needs to be approved before a user can translate this component.
Adding new translation:
Create new language file
How to handle requests for creating new translations.
Language code style:
Default based on the file format
Customize language code used to generate the filename for translations created by Weblate.
Manage strings
Enables adding and removing strings straight from Weblate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.
Merge style:
Rebase
Define whether Weblate should merge the upstream repository or rebase changes onto it.
Commit message when translating:
Translated using Weblate ({{ language_name }})
Currently translated at {{ stats.translated_percent }}% ({{ stats.translated }} of {{ stats.all }} strings)
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Commit message when adding translation:
Added translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when removing translation:
Deleted translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when merging translation:
Merge branch '{{ component_remote_branch }}' into Weblate.
You can use template language for various info, please consult the documentation for more details.
Commit message when add-on makes a change:
Update translation files
Updated by "{{ addon_name }}" hook in Weblate.
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Merge request message:
Translations update from {{ site_title }}
Translations update from [{{ site_title }}]({{ site_url }}) for [{{ project_name }}/{{ component_name }}]({{url}}).
{% if component_linked_childs %}
It also includes following components:
{% for linked in component_linked_childs %}
* [{{ linked.project_name }}/{{ linked.name }}]({{ linked.url }})
{% endfor %}
You can use template language for various info, please consult the documentation for more details.
Push on commit
Whether the repository should be pushed upstream on every commit.
Age of changes to commit:
24
Time in hours after which any pending changes will be committed to the VCS.
Lock on error
Whether the component should be locked on repository errors.
Source language:
English
Language used for source strings in all components
Language filter:
^(xx|hu|cs)$
Regular expression used to filter translation files when scanning for file mask.
Variants regular expression:
Regular expression used to determine variants of a string.
Priority:
Medium
Components with higher priority are offered first to translators.
Restricted component
Restrict access to the component to only those explicitly given permission.
Share in projects:
WeblateOrg
Choose additional projects where this component will be listed. Hold down "Control", or "Command" on a Mac, to select more than one.
Use as a glossary
Glossary color:
Silver
Remote revision:
Local revision:
SAVE
Save and add another
Save and continue editing

Ver también:

Configuración de componentes, *Formatos bilingües y monolingües*

Componentes monolingües

Para facilitar la traducción de los mismos, proporciona un archivo de plantilla que contenga la asignación de los ID de los mensajes a su respectivo idioma de origen (normalmente el inglés). (Véase *Configuración de componentes* para obtener información sobre los parámetros individuales):

Weblate administration
WELCOME, WEBLATE TEST. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Home › Weblate translations › Components › Add Component
Add Component
IMPORT GPEED DOCUMENTATION
Required fields are marked in bold.
Component name:
Android
Display name
URL slug:
android
Name used in URLs and filenames.
Project:
WeblateOrg
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:
weblate://weblateorg/language-names
URL of a repository, use weblate://project/component to share it with other component.
Repository push URL:
URL of a push repository, pushing is turned off if empty.
Repository browser:
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}}.
Exported repository URL:
URL of repository where users can fetch changes from Weblate.
Source string bug reporting address:
E-mail address for reports on errors in source strings. Leave empty for no e-mails.
Repository branch:
Repository branch to translate.
Push branch:
Branch for pushing changes, leave empty to use repository branch
File mask:
app/src/main/res/values-*/strings.xml
Path of files to translate relative to repository root, use * instead of language code, for example: po/* po or locale/*/LC_MESSAGES/djangojs.po.
Screenshot file mask:
Path of screenshots relative to repository root, for example: docs/screenshots/*.png.
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.
Template for new translations:
Filename of file used for creating new translations. For gettext choose .pot file.
File format:
Android String Resource
Locked
Locked component will not get any translation updates.
Allow translation propagation
Whether translation updates in other components will cause automatic translation in this one
Turn on suggestions
Whether to allow translation suggestions at all.
Suggestion voting
Users can only vote for suggestions and can't make direct translations.
Autoaccept suggestions:
0
Automatically accept suggestions with this number of votes, use 0 to turn it off.
Translation flags:
Additional comma-separated flags to influence Weblate behavior.
Enforced checks:
[]
List of checks which can not be ignored.
Translation license:
MIT License
Contributor agreement:
User agreement which needs to be approved before a user can translate this component.
Adding new translation:
Create new language file
How to handle requests for creating new translations.
Language code style:
Default based on the file format
Customize language code used to generate the filename for translations created by Weblate.
Manage strings
Enables adding and removing strings straight from Weblate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.
Merge style:
Rebase
Define whether Weblate should merge the upstream repository or rebase changes onto it.
Commit message when translating:
Translated using Weblate ({{ language_name }})
Currently translated at {{ stats.translated_percent }}% ({{ stats.translated }} of {{ stats.all }} strings)
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Commit message when adding translation:
Added translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when removing translation:
Deleted translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when merging translation:
Merge branch '{{ component_remote_branch }}' into Weblate.
You can use template language for various info, please consult the documentation for more details.
Commit message when add-on makes a change:
Update translation files
Updated by "{{ addon_name }}" hook in Weblate.
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Merge request message:
Translations update from {{ site_title }}
Translations update from [{{ site_title }}]({{ site_url }}) for [{{ project_name }}/{{ component_name }}]({{url}}).
{% if component_linked_childs %}
It also includes following components:
{% for linked in component_linked_childs %}
* [{{ linked.project_name }}/{{ linked.name }}]({{ linked.url }})
{% endfor %}
You can use template language for various info, please consult the documentation for more details.
Push on commit
Whether the repository should be pushed upstream on every commit.
Age of changes to commit:
24
Time in hours after which any pending changes will be committed to the VCS.
Lock on error
Whether the component should be locked on repository errors.
Source language:
English
Language used for source strings in all components
Language filter:
^[^.]+$
Regular expression used to filter translation files when scanning for file mask.
Variants regular expression:
Regular expression used to determine variants of a string.
Priority:
Medium
Components with higher priority are offered first to translators.
Restricted component
Restrict access to the component to only those explicitly given permission.
Share in projects:
WeblateOrg
Choose additional projects where this component will be listed. Hold down "Control", or "Command" on a Mac, to select more than one.
Use as a glossary
Glossary color:
Silver
Remote revision:
Local revision:
SAVE
Save and add another
Save and continue editing

Ver también:

Configuración de componentes, *Formatos bilingües y monolingües*

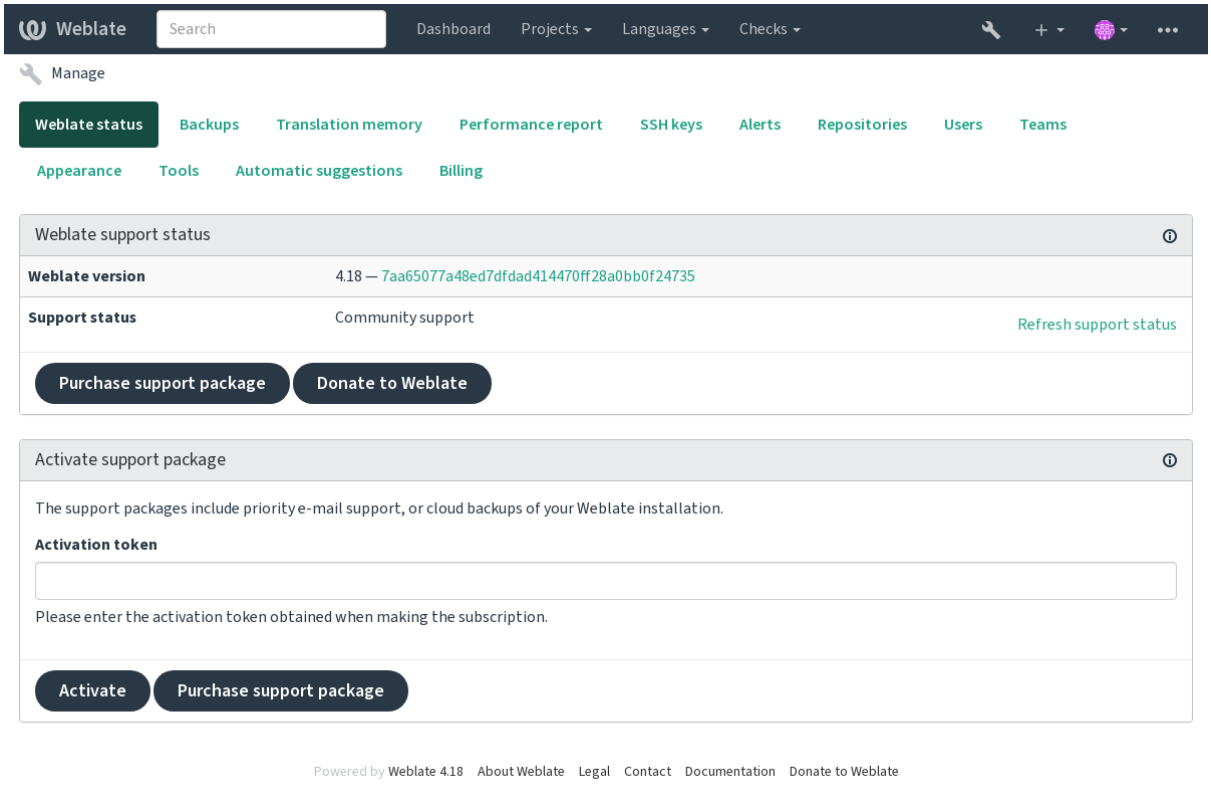
2.24 Obtener ayuda con Weblate

Weblate es *software* libre, de licencia *copyleft*, con asistencia comunitaria. Los suscriptores reciben asistencia prioritaria sin ningún costo adicional. Hay paquetes de ayuda prepagados disponibles para todos. Hallará más información relativa a las opciones de asistencia actuales en <https://weblate.org/es/support/>.

2.24.1 Integrating support

Nuevo en la versión 3.8.

Purchased support packages can optionally be integrated into your Weblate subscription management interface, from where you will find a link to it. Basic instance details about your installation are also reported back to Weblate this way.



2.24.2 Data submitted to the Weblate

- URL en la que se ha configurado su instalación de Weblate
- El título de su sitio
- La versión de Weblate que ejecuta
- Tallies of some objects in your Weblate database (projects, components, languages, source strings and users)
- La clave pública de SSH de su instalación

Additionally, when *Descubrir Weblate* is turned on:

- List of public projects (name, URL and website)

No se envía ningún otro dato.

2.24.3 Integration services

- See if your support package is still valid
- *Almacenamiento de copia de seguridad proporcionado por Weblate*
- *Descubrir Weblate*

Consejo: Purchased support packages are already activated upon purchase, and can be used without integrating them.

2.24.4 Descubrir Weblate

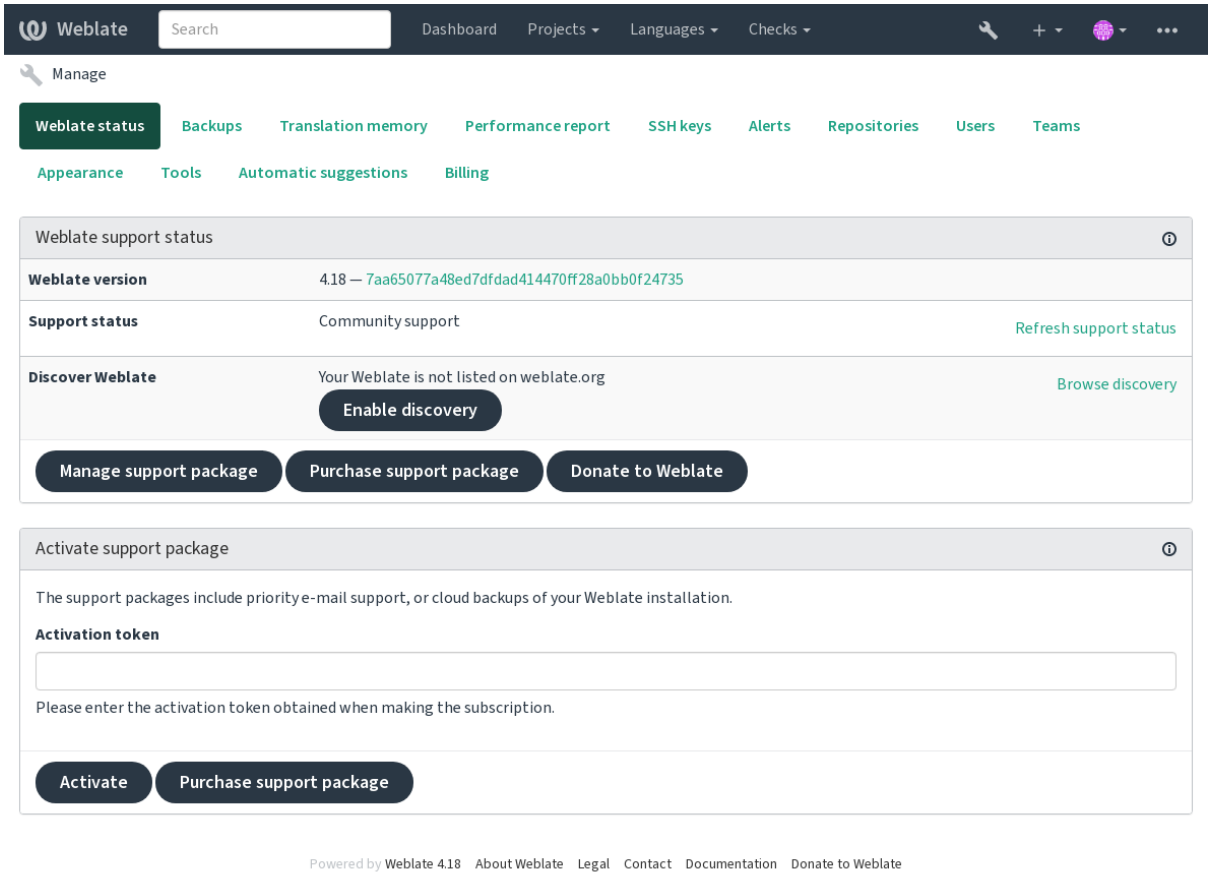
Nuevo en la versión 4.5.2.

Discover Weblate is an opt-in service that makes it easier for users to find Weblate servers and communities. Users can browse registered services on <https://weblate.org/discover/>, and find there projects to contribute.

Cómo aparece en la lista

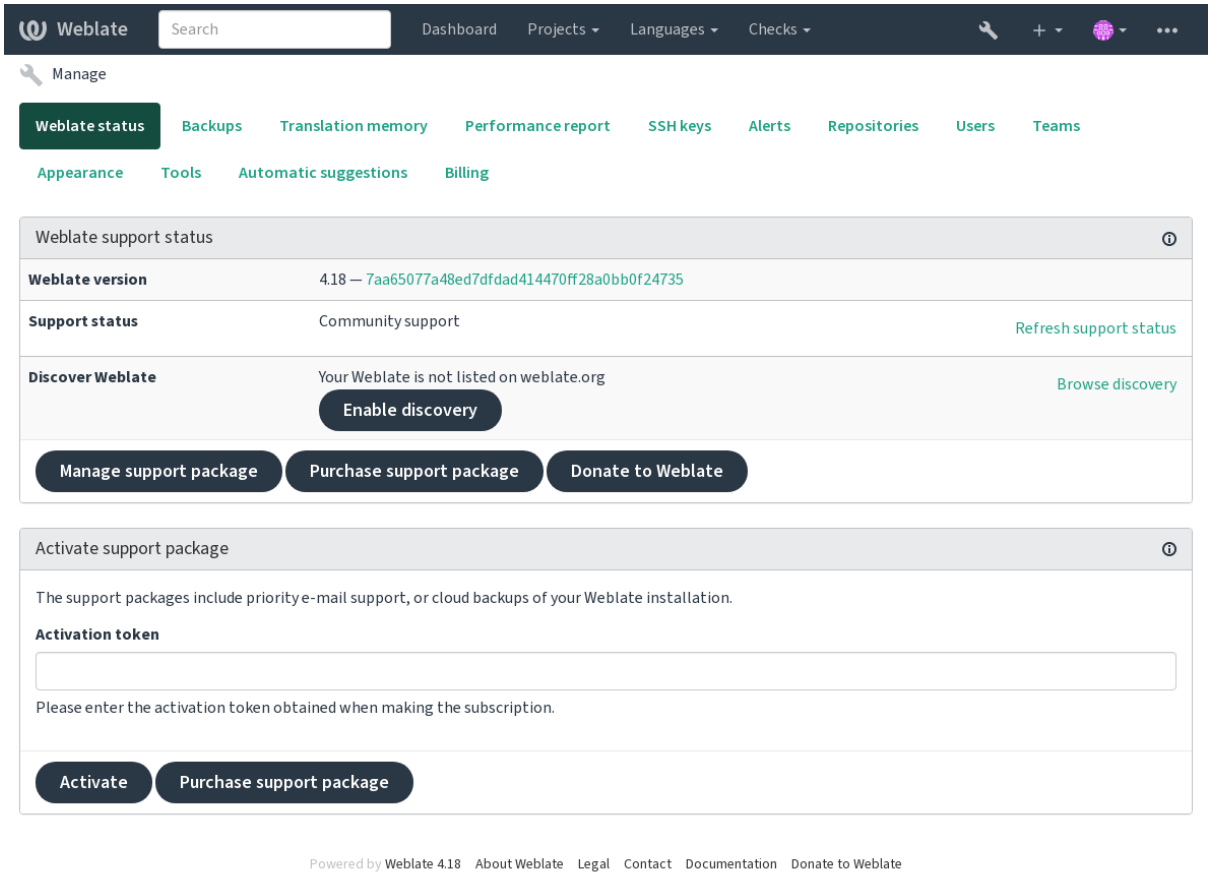
Consejo: Participating in Discover Weblate makes Weblate submit some information about your server, please see *Data submitted to the Weblate*.

To list your server with an active support subscription (see *Integrating support*) in Discover Weblate all you need to do is turn this on in the management panel:



Listing your server without a support subscription in Discover Weblate:

1. Regístrese en <https://weblate.org/user/>
2. Register your Weblate server in the discovery database at <https://weblate.org/subscription/discovery/>
3. Confirm the service activation in your Weblate and turn on the discovery listing in your Weblate management page using *Enable discovery* button:



Personalización de la lista

You can customize the listing by providing a text and image (570 x 260 pixels) at <https://weblate.org/user/>.

2.25 Documentos legales

Nota: Herein you will find various legal information you might need to operate Weblate in certain legal jurisdictions. It is provided as a means of guidance, without any warranty of accuracy or correctness. It is ultimately your responsibility to ensure that your use of Weblate complies with all applicable laws and regulations.

2.25.1 Licensing compliance

Weblate comes with REUSE 3.0 compliant license specification.

2.25.2 ITAR and other export controls

Weblate can be run within your own datacenter or virtual private cloud. As such, it can be used to store ITAR or other export-controlled information, however, end users are responsible for ensuring such compliance.

The Hosted Weblate service has not been audited for compliance with ITAR or other export controls, and does not currently offer the ability to restrict translations access by country.

2.25.3 US encryption controls

Weblate does not contain any cryptographic code, but might be subject export controls as it uses third party components utilizing cryptography for authentication, data-integrity and -confidentiality.

Most likely Weblate would be classified as ECCN 5D002 or 5D992 and, as publicly available libre software, it should not be subject to EAR (see Encryption items NOT Subject to the EAR).

Software components used by Weblate (listing only components related to cryptographic function):

Python
: See https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG
: Optionally used by Weblate

Git
: Optionally used by Weblate

curl
: Used by Git

OpenSSL
: Used by Python and cURL

The strength of encryption keys depends on the configuration of Weblate and the third party components it interacts with, but in any decent setup it will include all export restricted cryptographic functions:

- In excess of 56 bits for a symmetric algorithm
- Factorisation of integers in excess of 512 bits for an asymmetric algorithm
- Computation of discrete logarithms in a multiplicative group of a finite field of size greater than 512 bits for an asymmetric algorithm
- Discrete logarithms in a group different than above in excess of 112 bits for an asymmetric algorithm

Weblate doesn't have any cryptographic activation feature, but it can be configured in a way where no cryptography code would be involved. The cryptographic features include:

- Accessing remote servers using secure protocols (HTTPS)
- Generating signatures for code commits (PGP)

Ver también:

Export Controls (EAR) on Open Source Software

CAPÍTULO 3

Documentación para contribuidores

3.1 Contribuir con Weblate

There are dozens of ways to improve Weblate. You can choose the one you feel comfortable with, be it coding, graphics design, documentation, sponsorship, or an idea:

- *Informar de problemas en Weblate*
- *Empezar a contribuir con código a Weblate*
- *Contributing to Weblate modules*
- *Traducir Weblate*
- *Contribute to Weblate documentation*
- *Weblate discussions*
- *Funding Weblate development*

3.1.1 Traducir Weblate

Weblate is continually being translated using Weblate itself. Feel free to take your part in the effort of making Weblate available in as many human languages as possible. It brings Weblate closer to its users!

If you find a possible mistake in the source string, you can mark it with a comment in the Weblate editor. This way, it can be discussed and corrected. If you're certain, you can also click on the link in the *Source string location* section and submit a PR with your correction.

3.1.2 Contribute to Weblate documentation

You are welcome to improve the documentation page of your choice. Do it easily by clicking the *Edit on GitHub* button in the top-right corner of the page.

Please respect these guidelines while writing:

1. Don't remove part of the documentation if it's valid.
2. Use clear and easily-understandable language. You are writing tech docs, not a poem. Not all docs readers are native speakers, be thoughtful.
3. Don't be afraid to ask if you are not certain. If you have to ask about some feature while editing, don't change its docs before you have the answer. This means: You change or ask. Don't do both at the same time.
4. Verify your changes by performing described actions while following the docs.
5. Send PR with changes in small chunks to make it easier and quicker to review and merge.
6. If you want to rewrite and change the structure of a big article, do it in two steps:
    1. Rewrite
    2. Once the rewrite is reviewed, polished, and merged, change the structure of the paragraphs in another PR.

Consejo: You can translate the docs.

3.1.3 Extending built-in language definitions

The language definitions are in the weblate-language-data repository.

You are welcome to add missing language definitions to `languages.csv`, other files are generated from that file.

3.1.4 Weblate discussions

If you have an idea and not sure if it's suitable for an issue, don't worry. You can join the community in GitHub discussions.

3.1.5 Funding Weblate development

You can boost Weblate's development on the donate page. Funds collected there are used to enable gratis hosting for libre software projects and further development of Weblate. Please check the donate page for options, such as funding goals and the rewards you get as a proud funder.

Partidarios que han financiado Weblate

List of Weblate supporters:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- Cassidy James
- Loic Dachary
- Marozed

- https://freedombox.org/es/
- GNU Solidario (GNU Health)
- BallotReady
- Richard Nespithal
- MyExpenses.Mobi
- Michael Totschnig
- James Valleroy

Do you want to be in the list? Please see options on the Donate to Weblate.

3.2 Empezar a contribuir con código a Weblate

Para entender el código fuente de Weblate, explore *Código fuente de Weblate*, *Interfaz de Weblate* y *Weblate por dentro*.

3.2.1 Primeros pasos con el código

Familiarize yourself with the Weblate codebase, by having a go at the bugs labelled good first issue.

You are welcome to start working on these issues without asking. Just announce that in the issue, so that it's clear that somebody is working on that issue.

3.2.2 Ejecutar Weblate localmente

The most comfortable approach to get started with Weblate development is to follow *Instalar desde el código fuente*. It will get you a virtualenv with editable Weblate sources.

1. Clone el código fuente de Weblate:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Cree un entorno virtual:

```
virtualenv .venv
. .venv/bin/activate
```

3. Instale Weblate (para esto necesitará algunas dependencias de sistema; vea *Instalar desde el código fuente*):

```
pip install -e '.[all]'
```

3. Install all dependencies useful for development:

```
pip install -r requirements-dev.txt
```

4. Inicie un servidor de desarrollo:

```
weblate runserver
```

5. Depending on your configuration, you might also want to start Celery workers:

```
./weblate/examples/celery start
```

6. To run a test (see *Puesta a prueba local* for more details):

```
. scripts/test-database.sh
./manage.py test
```

Ver también:

Instalar desde el código fuente

3.2.3 Ejecutar Weblate localmente en Docker

If you have Docker and docker-compose-plugin 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 *Añadir proyectos y componentes de traducción*.

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
```

Nota: Be careful that your Docker containers are up and running before running the tests. You can check that by running the `docker ps` command.

Para mostrar los registros:

```
./rundev.sh logs
```

Para detener los contenedores en segundo plano, ejecute:

```
./rundev.sh stop
```

Ejecutar la secuencia de órdenes sin argumentos recreará el contenedor de Docker y lo reiniciará.

Nota: Esta no es una puesta en funcionamiento ideal para entornos de producción, puesto que incluye varios arreglos apurados inseguros que, sin embargo, facilitan el desarrollo.

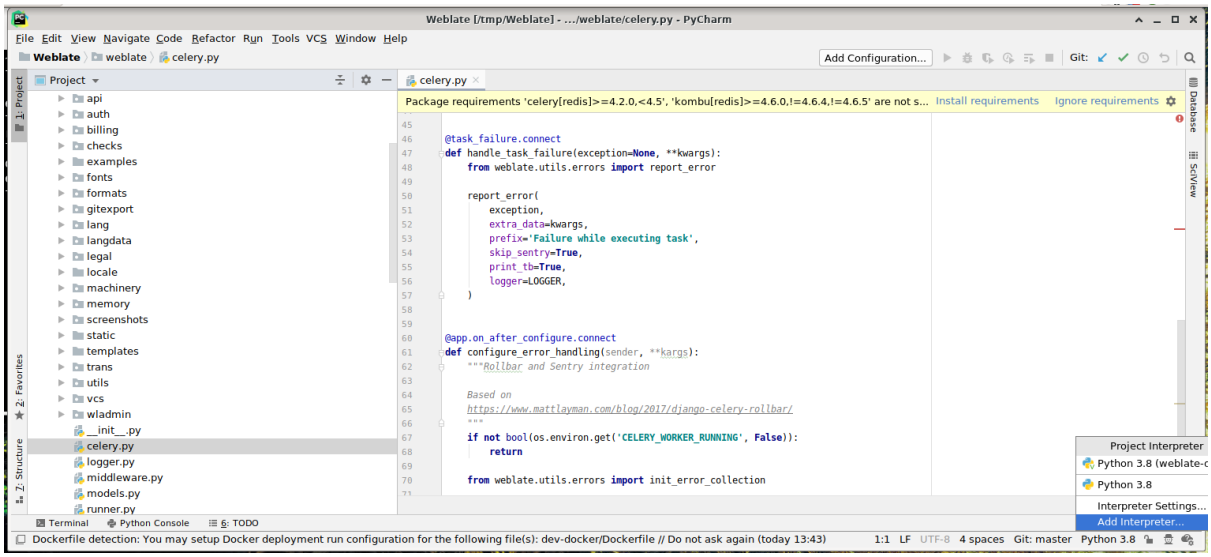
3.2.4 Bootstrapping your devel instance

You might want to use *`weblate weblate import_demo`* to create demo translations and *`weblate weblate createadmin`* to make an admin user.

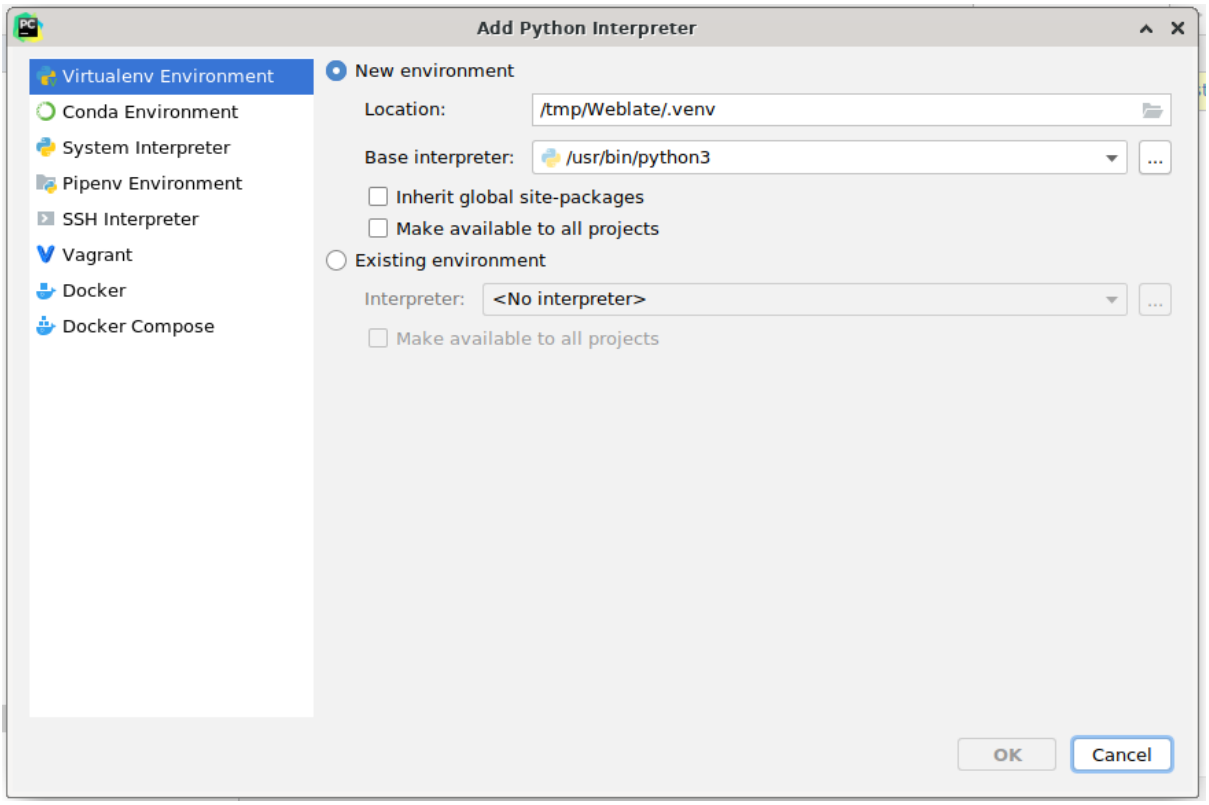
3.2.5 Trabajar en Weblate con PyCharm

PyCharm es un conocido EID para Python. A continuación le ofrecemos algunas recomendaciones para configurar Weblate en este programa.

Considering you have just cloned the GitHub repository to a folder, just open it with PyCharm. Once the IDE is open, the first step is to specify the interpreter you want to use:



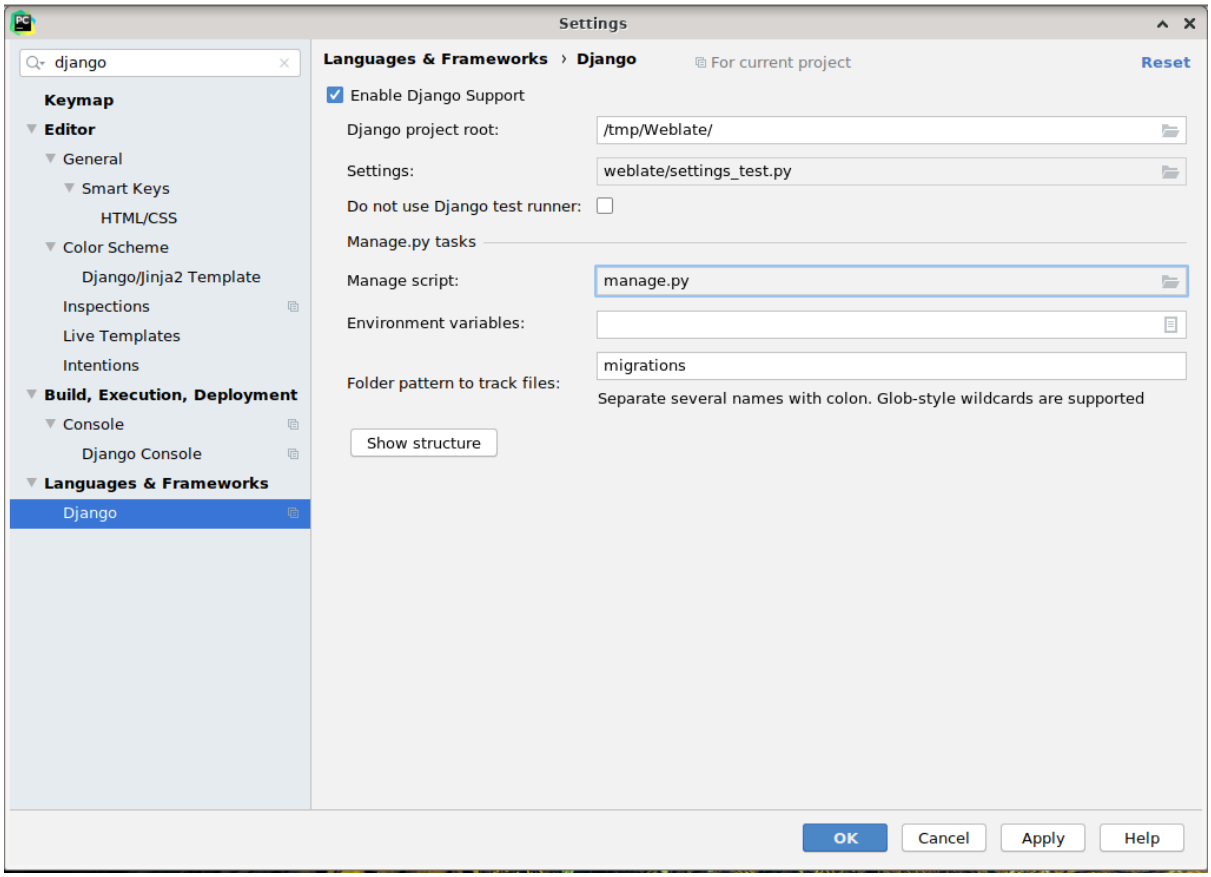
You can either choose to let PyCharm create the virtualenv for you, or select an already existing one:



Don't forget to install the dependencies once the interpreter is set: Either through the console (the console from the IDE will directly use your virtualenv by default), or through the interface when you get a warning about missing dependencies.

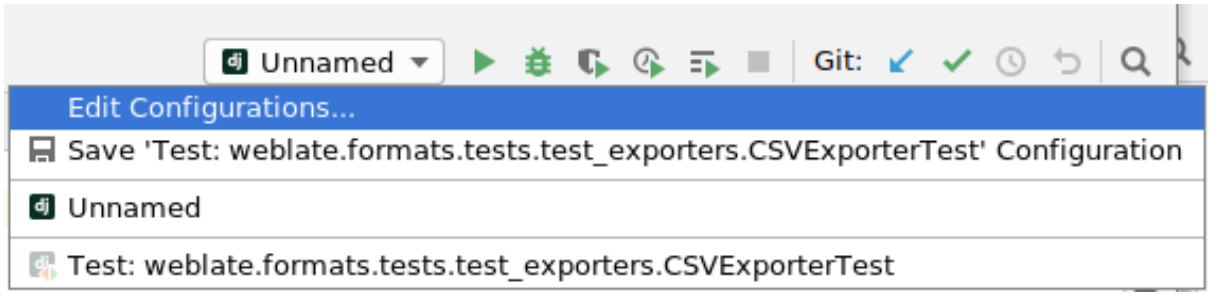
The second step is to set the right info to use Django natively inside PyCharm: The idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of the Django project and the path to its

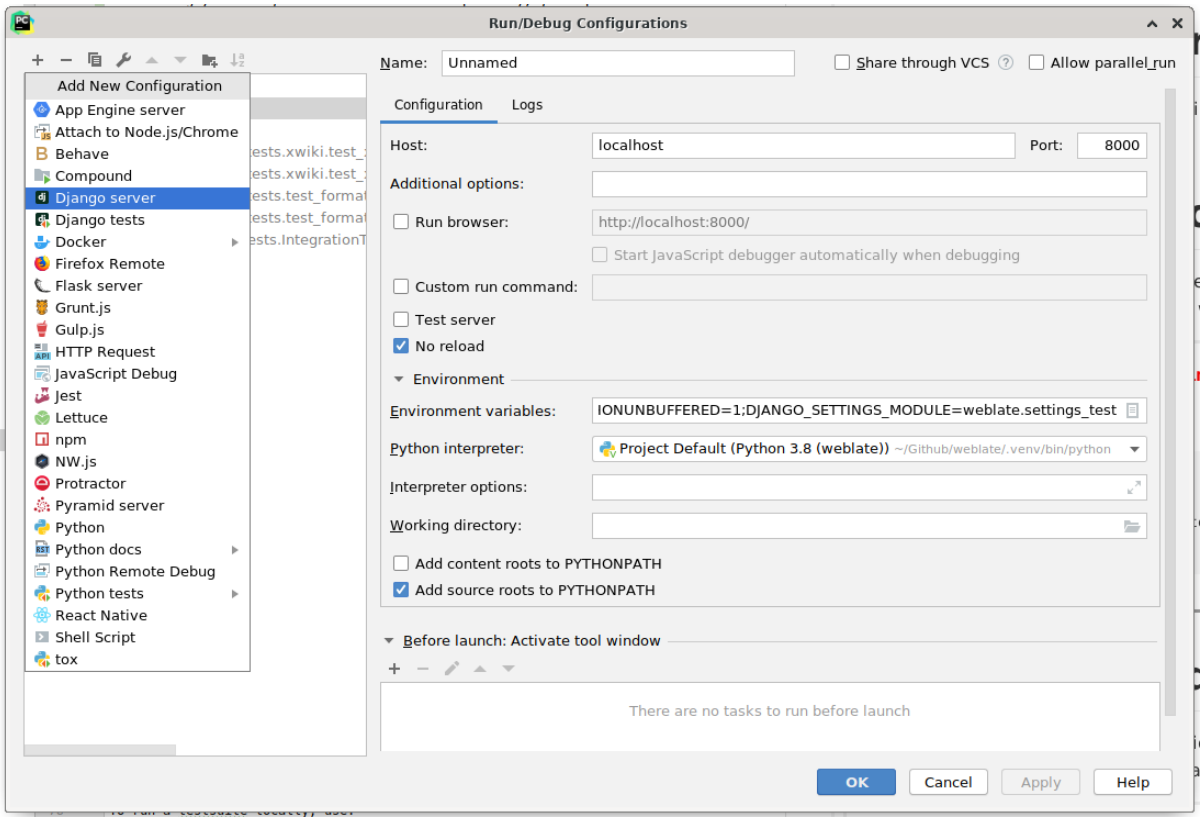
settings:



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:





Consejo: 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 Código fuente de Weblate

Weblate se desarrolla en GitHub. Le invitamos a copiar el código y enviarnos solicitudes de incorporación de cambios. También recibimos modificaciones por otras vías, como en forma de parches.

Ver también:

Check out *Weblate por dentro* to see how Weblate looks from inside.

3.3.1 Coding guidelines

Cualquier código escrito para Weblate debe crearse teniendo en mente los principios de seguridad por naturaleza.

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 Depurar 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 Modo de depuración

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.

Ver también:

Desactivar el modo de depuración, *Properly configure admins*, *Collecting error reports*

3.4.2 Registros de Weblate

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 *Tareas en segundo plano con Celery*) usually produces its own logs as well. The example system-wide setups logs to several files under `/var/log/celery/`.

Docker containers log to their output (as per usual in the Docker world), so you can look at the logs using `docker compose logs`. You can get more detailed logs by changing *WEBLATE_LOGLEVEL*.

Ver también:

Configuración de muestra 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.

Elementos que revisar en este caso:

- Check that the Celery process is running, see *Tareas en segundo plano con Celery*
- Check the Celery queue status, either in *Interfaz de gestión*, or using `weblate weblate celery_queues`
- Look in the Celery logs for errors (see *Registros de Weblate*)

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 *Invocar órdenes de gestión* for instructions on how to invoke it in different environments) or by using *Interfaz de gestión* under the *Tools* tab.

These send e-mails directly, so this verifies that your SMTP configuration is correct (see *Configurar el correo electrónico saliente*). 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 *Collecting error reports*.

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 *Collecting error reports* helps you to notice such failures easier.

3.4.7 Problemas de rendimiento

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 *Collecting error reports* 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 por dentro

Nota: This chapter will give you basic overview of Weblate internals.

Weblate derives most of its code structure from, and is based on Django.

3.5.1 Estructura de directorios

Quick overview of directory structure of Weblate main repository:

docs
Source code for this documentation, which can be built using Sphinx.

dev-docker
Docker code to run development server, see *Ejecutar Weblate localmente en Docker*.

weblate
Source code of Weblate as a Django application, see *Weblate por dentro*.

weblate/static
Client files (CSS, Javascript and images), see *Interfaz de Weblate*.

3.5.2 Módulos

Weblate se compone de varias aplicaciones de Django (algunas son opcionales; vea *Módulos opcionales de Weblate*):

`accounts`

Cuenta de usuario, perfiles y notificaciones.

`addons`

Add-ons to tweak Weblate behavior, see *Complementos*.

`api`

API based on Django REST framework.

`auth`

Authentication and permissions.

`billing`

The optional *Facturación* module.

`checks`

Translation string *Comprobaciones de calidad* module.

`fonts`

Font rendering checks module.

`formats`

File format abstraction layer based on translate-toolkit.

`gitexport`

The optional *Git exporter* module.

`lang`

Módulo que define los idiomas y los modelos de pluralización.

`legal`

El módulo facultativo *Información legal*.

`machinery`

Integración de servicios de traducción automática.

`memory`

Memoria de traducción integrada, consulte *Memoria de traducción*.

`screenshots`

Gestión de capturas de pantalla y módulo de OCR.

`trans`

Módulo principal que manipula las traducciones.

`utils`

Diversas utilidades auxiliares.

`vcs`

Abstracción del sistema de control de versiones.

`wladmin`

Personalización de la interfaz administrativa de Django.

3.6 Developing add-ons

Complementos are way to customize localization workflow in Weblate.

class `weblate.addons.base.`**BaseAddon**(*storage=None*)

Base class for Weblate add-ons.

classmethod can_install(*component*, *user*)

Check whether add-on is compatible with given component.

configure(*settings*)

Guardar configuración.

daily(*component*)

Actuador desencadenado diariamente.

classmethod get_add_form(*user*, *component*, ***kwargs*)

Return configuration form for adding new add-on.

get_settings_form(*user*, ***kwargs*)

Return configuration form for this add-on.

post_add(*translation*)

Actuador desencadenado tras añadirse una traducción nueva.

post_commit(*component*)

Actuador desencadenado tras consignarse cambios en el repositorio.

post_push(*component*)

Actuador desencadenado tras enviar el repositorio a la ubicación del proyecto originario.

post_update(*component*, *previous_head: str*, *skip_push: bool*)

Actuador desencadenado tras actualizarse el repositorio desde el origen.

Parámetros

- **previous_head** (*str*) – «HEAD» del repositorio antes de la actualización; puede estar vacía durante la clonación inicial.
- **skip_push** (*bool*) – Whether the add-on operation should skip pushing changes upstream. Usually you can pass this to underlying methods as `commit_and_push` or `commit_pending`.

pre_commit (*translation*, *author*)

El enlace se activa antes de que los cambios se confirmen en el repositorio.

pre_push (*component*)

Hook triggered before repository is pushed upstream.

pre_update (*component*)

Hook triggered before repository is updated from upstream.

save_state ()

Guarde la información del estado del complemento.

store_post_load (*translation*, *store*)

Actuador desencadenado tras procesar un archivo.

It receives an instance of a file format class as a argument.

This is useful to modify file format class parameters, for example adjust how the file will be saved.

unit_pre_create (*unit*)

Actuador desencadenado antes de crearse una unidad nueva.

Here is an example add-on:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy

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 = gettext_lazy("Example add-on")
    # Detailed add-on description
    description = gettext_lazy("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Interfaz de Weblate

La interfaz está elaborada con Bootstrap, jQuery y algunas bibliotecas de terceros.

3.7.1 Navegadores admitidos

Weblate es compatible con las versiones estables más recientes de todos los navegadores web principales, en todas las plataformas.

No se brinda servicio técnico explícito al emplear navegadores alternativos que hacen uso (directo o a través de la API de visualización web de la plataforma) de la versión más reciente de WebKit, Blink o Gecko. Sin embargo, Weblate se visualizará y funcionará correctamente en esos navegadores, en la mayoría de los casos.

Es posible que navegadores más antiguos funcionen, pero determinadas funciones se verán limitadas.

3.7.2 Gestión de dependencias

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 Coding style

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 Localización

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 = ngettext('literal for the singular case',
        'literal for the plural case', object_count);

fmts = ngettext('There is %s object. Remaining: %s',
        'There are %s objects. Remaining: %s', 11);
s = interpolate(fmts, [11, 20]);
// s is 'There are 11 objects. Remaining: 20'
```

Ver también:

Translation topic in the Django documentation

3.7.5 Iconos

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 Informar de problemas en Weblate

El seguimiento de errores de Weblate se aloja en GitHub.

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 *Problemas de seguridad* section below.

If you are not sure about your bug report or feature request, you can try *Weblate discussions*.

3.8.1 Problemas de seguridad

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.

Nota: 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.

Some of these are:

- Django
- Django REST framework
- 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 Integración continua

Current test results can be found on GitHub Actions and coverage is reported on Codecov.

There are several jobs to verify different aspects:

- Pruebas unitarias
- 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 Puesta a prueba local

Para ejecutar un conjunto de pruebas localmente, utilice:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Consejo: 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 *Configuración de base de datos para Weblate* for setup instructions.

The `weblate/settings_test.py` is used in CI environment as well (see *Integración continua*) 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
```

También puede especificar que se ejecuten solo pruebas individuales:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Consejo: The tests can also be executed inside developer docker container, see *Ejecutar Weblate localmente en Docker*.

Ver también:

See Testing in Django for more info on running and writing tests for Django.

3.10 Esquemas de datos

Weblate uses JSON Schema to define layout of external JSON files.

3.10.1 Weblate Translation Memory Schema

https://weblate.org/schemas/weblate-memory.schema.json			
type	*array*		
items	*The Translation Memory Item*		
	type	*object*	
	properties		
	• **category**	*The String Category*	
		1 is global, 2 is shared, 10000000+ are project specific, 20000000+ are user specific	
		type	*integer*
		examples	1
		minimum	0
		default	1
	• **origin**	*The String Origin*	
		Filename or component name	

continué en la próxima página

Tabla 1 – proviene de la página anterior

		type	*string*
		examples	test.tmx
			proyecto/componente
		default	
	• **source**	*The Source String*	
		type	*string*
		examples	Hello
		minLength	1
		default	
	• **source_language**	*The Source Language*	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		type	*string*
		examples	es
		pauta	^[^]+$
		default	
	• **target**	*The Target String*	
		type	*string*
		examples	Ahoj
		minLength	1
		default	
	• **target_language**	*The Target Language*	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		type	*string*
		examples	cs
		pauta	^[^]+$
		default	
	additionalProperties	Falso	
definiciones			

Ver también:

Memoria de traducción, `weblate weblate dump_memory`, `weblate weblate import_memory`

3.10.2 Exportación de datos de usuarios de Weblate

https://weblate.org/schemas/weblate-userdata.schema.json			
type	*object*		
properties			
• **basic**	Básico		
	type	*object*	
	properties		
	• **username**	*Nombre de usuario*	
		type	*string*
		examples	administrador
		default	
	• **full_name**	*Nombre completo*	
		type	*string*
		examples	Weblate Admin
		default	
	• **email**	*Correo electrónico*	
		type	*string*
		examples	noreply@example.com
		default	
		format	email

continué en la próxima página

Tabla 2 – proviene de la página anterior

	• **date_joined**	*Fecha de incorporación*	
		type	*string*
		examples	2019-11-18T18:53:54.862Z
		default	
		format	date-time
	additionalProperties	Falso	
• **profile**	*Perfil*		
	type	*object*	
	properties		
	• **language**	*Idioma*	
		type	*string*
		examples	cs
		pauta	^[^]*$
		default	
	• **suggested**	*Number of suggested strings*	
		type	*integer*
		examples	1
		default	0
	• **translated**	*Number of translated strings*	
		type	*integer*
		examples	24
		default	0
	• **uploaded**	*Number of uploaded screenshots*	
		type	*integer*
		examples	1
		default	0
	• **hide_completed**	*Hide completed translations on the dashboard*	
		type	*boolean*
		examples	Falso
		default	Verdadero
	• **secondary_in_zen**	*Mostrar traducciones secundarias en modo zen*	
		type	*boolean*
		examples	Verdadero
		default	Verdadero
	• **hide_source_sec**	*Ocultar origen si existe traducción secundaria*	
		type	*boolean*
		examples	Falso
		default	Verdadero
	• **editor_link**	*Enlace del editor*	
		type	*string*
		examples	
		pauta	^.*$
		default	
	• **translate_mode**	*Modo del editor de traducciones*	
		type	*integer*
		examples	0
		default	0
	• **zen_mode**	*Modo de editor zen*	
		type	*integer*
		examples	0
		default	0
	• **special_chars**	*Caracteres especiales*	
		type	*string*
		examples	
		pauta	^.*$

continué en la próxima página

Tabla 2 – proviene de la página anterior

		default		
	• theme	*Theme*		
		type	*string*	
		examples	auto	
		pauta	^(auto|dark|light)$	
		default	auto	
	• **dashboard_view**	*Vista predeterminada del cuadro de mando*		
		type	*integer*	
		examples	1	
		default	0	
	• **dashboard_compo**	*Lista de componentes predeterminados*		
		default	null	
		anyOf	type	*null*
			type	*integer*
	• **languages**	*Idiomas traducidos*		
		type	*array*	
		default		
		items	*Código de idioma*	
			type	*string*
			examples	cs
			pauta	^.*$
			default	
	• **secondary_languag**	*Idiomas secundarios*		
		type	*array*	
		default		
		items	*Código de idioma*	
			type	*string*
			examples	sk
			pauta	^.*$
			default	
	• **mirado**	*Proyectos monitorizados*		
		type	*array*	
		default		
		items	*«Slug» del proyecto*	
			type	*string*
			examples	weblate
			pauta	^.*$
			default	
	additionalProperties	Falso		
• **registro de auditoría**	*Registro de auditoría*			
	type	*array*		
	default			
	items	*Elementos*		
		type	*object*	
		properties		
		• **address**	*Dirección IP*	
			type	*string*
			examples	127.0.0.1
			pauta	^.*$
			default	
		• **Agente_ de usuario**	*Agente de usuario*	
			type	*string*
			examples	PC / Linux / Firefox 70.0
			pauta	^.*$

continué en la próxima página

Tabla 2 – proviene de la página anterior

		default	
	• **fecha de inicio**	*Cronomarcador*	
		type	*string*
		examples	2019-11-18T18:58:30.845Z
		default	
		format	date-time
	• **actividad**	*Actividad*	
		type	*string*
		examples	acceso
		pauta	^.*$
		default	
	additionalProperties	Falso	
definiciones			

Ver también:

Perfil de usuario, `weblate weblate 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.

Ver también:

Actualizar 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 Seguridad y privacidad

Truco: Para Weblate, la seguridad es mantener un entorno que valore la privacidad de los usuarios.

Development of Weblate adheres to the Best Practices of the Linux Foundation's Core Infrastructure Initiative.

Ver también:

Problemas de seguridad

3.12.1 Security updates

Only the latest release is guaranteed to receive security updates.

3.12.2 Búsqueda de vulnerabilidades en las dependencias

Security issues in our dependencies are monitored using Dependabot. This covers the Python and JavaScript libraries, and the latest stable release has its dependencies updated to avoid vulnerabilities.

Consejo: There might be vulnerabilities in third-party libraries which do not affect Weblate, so those are not addressed by releasing bugfix versions of Weblate.

3.12.3 Docker container security

The Docker containers are regularly scanned using Anchore and Trivy security scanners.

This allows us to detect vulnerabilities early and release improvements quickly.

You can get the results of these scans at GitHub — they are stored as artifacts on our CI in the SARIF format (Static Analysis Results Interchange Format).

Ver también:

Integración continua

3.13 Contributing to Weblate modules

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 *Cliente de Weblate*
- translation-finder, used to discover translatable files in the repository
- language-data, definiciones de idioma para Weblate, consulte *Definiciones de idioma*

3.13.1 Coding guidelines

Cualquier código escrito para Weblate debe crearse teniendo en mente los principios de seguridad por naturaleza.

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
```

Ver también:

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
```

Ver también:

Código fuente de Weblate

3.14 Acerca de Weblate

3.14.1 Objetivos del proyecto

Web-based continuous localization tool with tight *Integración de control de versiones* supporting a wide range of *file formats*, making it easy for translators to contribute.

3.14.2 Nombre del proyecto

«Weblate» es una palabra híbrida creada a partir de las voces inglesas «web» y «translate».

3.14.3 Sitio web del proyecto

The landing page is https://weblate.org and there is a cloud-hosted service at https://hosted.weblate.org. The documentation can be read at https://docs.weblate.org.

3.14.4 Logotipos del proyecto

The project logos and other graphics are available in https://github.com/WeblateOrg/graphics.

3.14.5 Leadership

This project is maintained by Michal Čihař, who can be reached at michal@weblate.org.

3.14.6 Autores

Weblate was started by Michal Čihař. Since its inception in 2012, thousands of people have contributed.

3.15 Licencia

More detailed licensing information is available in the Weblate source code and follows REUSE 3.0 specification.

Copyright © Michal Čihař michal@weblate.org

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

CAPÍTULO 4

Historial de cambios

4.1 Weblate 4.18.2

Released on June 30th 2023.

- Fixed parsing notes from TBX.
- Fixed query parsing in navigation bar search.
- Fixed language filtering in reports.
- Improved ModernMT languages mapping.
- Disabled reused checks on languages with a single plural form.

All changes in detail.

4.2 Weblate 4.18.1

Released on June 16th 2023.

- Fixed language code format for i18next.
- Fixed CSS compression with dark theme.

All changes in detail.

4.3 Weblate 4.18

Released on June 15th 2023.

- Improved API error messages on permission denied.
- Reduced false positives of the XML checks.
- Translated check accuracy has been improved.
- Extended support for Fluent file format.
- Avoiding signing-out user in some rate-limits.

- Added support for storing glossary explanation in TBX format.
- Added support for ordering strings by last update.
- Extended search capabilities for finding users.
- Support for automatic update of screenshots from a repository.
- Improved translation memory performance.
- Project stats exports to JSON/CSV now include more details; it now matches content available in the API.
- Added check for reused translation.
- Highlight suggested change in automatic suggestions.
- Added dark theme; browser-following and manual setting are available.
- The Docker container can now be used with read-only root filesystem.
- Added username autocompletion when adding users to a project.
- Added site-wide search for projects, components, languages and users.
- New add-on Fill read-only strings with source.

All changes in detail.

4.4 Weblate 4.17

Released on April 17th 2023.

- Allow to filter on language in reports.
- Dropped deprecated command `cleanup_celery`.
- Fixed private project visibility for some teams.
- Automatic translation now honors target state when translating from other components.
- Improved performance of public user profiles.
- Improved Sentry performance integration.
- Added support for Ed25519 SSH key.
- Rewritten metrics storage.
- Added support for searching strings by position.
- Documentation improvements.
- Unchanged translation check can honor untranslatable terms from glossary.
- Added automatic fixup for Devanagari danda.
- Allow downloading project translation memory per language.
- Added new alert on unused components.

All changes in detail.

4.5 Weblate 4.16.4

Released on March 16th 2023.

- Dependencies updates.
- Improved background tasks scheduling.

All changes in detail.

4.6 Weblate 4.16.3

Released on March 15th 2023.

- Improved session handling with project backups.
- Dependencies updates.
- Actualización de las regionalizaciones.
- Documentation improvements.

All changes in detail.

4.7 Weblate 4.16.2

Released on March 8th 2023.

- Fixed searching in the translation memory.
- Fixed automatic translation with more services.
- Improved rendering of overlapping glossary term matches.
- Fixed plurals parsing for non-English source language in some formats.
- Added support for go-i18n v2 JSON files.

All changes in detail.

4.8 Weblate 4.16.1

Released on March 1st 2023.

- Fixed testsuite error.

All changes in detail.

4.9 Weblate 4.16

Released on March 1st 2023.

- Format string checks now also detects duplicated formats.
- Improved search performance for some specially formatted strings.
- Celery beat is now storing the tasks schedule in the database.
- Added support for IBM Watson Language Translator.
- Dropped support for VCS integration settings deprecated in 4.14.

- Added support for Bitbucket Server pull requests.
- Improved conflicts handling in gettext PO files.
- Added support for defining strings state when adding via API.
- Added support for configuring CORS allowed origins.
- Added plurals support to automatic suggestions.

All changes in detail.

4.10 Weblate 4.15.2

Released on January 25th 2023.

- Enabled gotext JSON and i18next v4 formats in the default configuration.
- Fixed crash on uploading corrupted files.
- Show stale directories in Git repository status.

All changes in detail.

4.11 Weblate 4.15.1

Released on January 19th 2023.

- Fixed suggestions from automatic translation.
- Fixed add-on page crash in some corner cases.
- Fixed untranslating template for new translations in some cases.
- Documented licensing using REUSE 3.0.
- Fixed users pagination on team management.
- Improved performance of project creation and saving.
- Added support for gotext JSON files.
- Added support for i18next v4 files.
- Pagination in the API is now customizable.

All changes in detail.

4.12 Weblate 4.15

Released on December 16th 2022.

- Se ha añadido un soporte para navegar por los cambios de una cadena individual.
- Se ha corregido el tratamiento de los plurales en la traducción automática desde otros componentes.
- Added keyboard shortcut Alt+Enter to submit string as a suggestion.
- Se ha añadido compatibilidad con las tablas de posición en el formato Fluent.
- Mejora del rendimiento de la memoria de traducción.
- Autogenerate repoweb browsing links for well known code hosting services.
- Improved performance of several views.

- Improved listing of strings with plurals.
- Added support for adding custom markup to HTML head.
- Fixed generation of MO files in the add-on to include only translated files.
- Fixed rendering of regular expression flags.
- Improved placeholders check behavior with plurals.
- Added support for translation files naming suitable for Google Play.
- Added support for labels in API.
- Added support for choosing different e-mail for commits than for notifications.
- The Docker image no longer enables debug mode by default.
- Order glossary terms based on the glossary component priority.
- Added team administrators who can add or remove members of the team.
- Added a popup confirmation before deleting users.
- Added add-on to customize XML output.

All changes in detail.

4.13 Weblate 4.14.2

Publicado el 5 de noviembre de 2022..

- Se ha añadido soporte para eliminar las entradas de la memoria de la traducción.
- Se ha mejorado el análisis en la alerta de los idiomas duplicados.
- Improved accuracy of the consecutive duplicated words check.
- Improved scaling of sending many notifications.
- Improved string state handling for subtitle translation.
- Deprecated insecure configuration of VCS service API keys via _TOKEN/_USERNAME configuration instead of _CREDENTIALS list.
- Fixed processing of some uploaded CSV files.
- Improved whitespace changes handling in diff display.
- Added automatic suggestions management link to management pages.
- Track comment removal/resolving in history.
- Fixed restoring project backups with linked components.
- Fixed captcha entering on unsuccessful registration.
- Improved languages support in DeepL.
- Improved webhooks compatibility with authenticated repositories.
- Added support for Python 3.11.

All changes in detail.

4.14 Weblate 4.14.1

Released on September 15th 2022.

- Fixed generating project backups in some situations.
- Improved error reporting on file upload.
- Fetch all user verified e-mails from GitHub during authentication.
- Avoid matching glossary terms on context or keys.
- Dispone de notificaciones para eliminaciones de cadenas.
- Se mejoró la gestión de los términos no traducibles en el glosario.
- List number of team members on team management page.
- Añadir una interfaz para la gestión de los grupos.
- Always show review stats when reviews are enabled.
- Added searching support in units API.
- Fixed progress bar display for read-only strings in the review workflow.
- Improved Burmese punctuation check.
- Fixed garbage collecting of metrics data.

All changes in detail.

4.15 Weblate 4.14

Publicada el 22 de agosto de 2022.

- Seguimiento de cambios de complementos en el historial.
- Fixed parsing translation from Windows RC, HTML and text files.
- Extended language code style configuration options.
- Added support for plurals updated in the recent CLDR releases.
- Se redujo el uso de memoria durante la actualización de componentes con muchas traducciones.
- Se permite el uso de dominios de traducción en SAP Translation Hub.
- Se permiten los enlaces absolutos en las ubicaciones de cadenas de origen.
- Improved operation behind some reverse proxies.
- Se amplió la API para abarcar la memoria de traducción.
- Se mejoró el flujo de traducción de documentos.
- Aumentó la fiabilidad de la traducción de archivos HTML y de texto.
- Se permite efectuar copias de respaldo a nivel proyecto.
- Se mejoró el desempeño y el uso de memoria de las búsquedas en la memoria de traducción.

Todos los cambios a detalle.

4.16 Weblate 4.13.1

Publicada el 1 de julio de 2022.

- Fixed tracking suggestions in history.
- Fixed parsing reverse proxy info from Cloudflare.
- Make parse error lock a component from translating.
- Fixed configuring intermediate file in the discovery add-on.
- Fixed DeepL translations behavior with placeholders.
- Fixed untranslating strings via API.
- Added support for removing user from a group via API.
- Fixed audit log for user invitation e-mails.
- Fixed flag names for Java formatting strings.

All changes in detail.

4.17 Weblate 4.13

Released on June 15th 2022.

- Se cambió el comportamiento de la actualización de nombres de idioma.
- Se añadió paginación a la lista de proyectos.
- La API para crear unidades nuevas ahora devuelve información sobre las unidades recién creadas.
- La detección de componentes ahora admite la configuración de un idioma intermedio.
- Added fixed encoding variants to CSV formats.
- Changed handling of context and location for some formats to better fit underlying implementation.
- Added support for ResourceDictionary format.
- Improved progress bar colors for color blind.
- Fixed variants cleanup on string removal.
- Compatibility with Django 4.1.
- Added support for storing escaped XML elements in XLIFF.
- Improved formatting of placeholder check errors.
- Redirect /.well-known/change-password to /accounts/password/.
- Machine translation services are now configurable per project.
- Added separate permission for resolving comments and grant it to the *Review strings* role.
- Added support for storing alternative translations in the CSV file.
- The placeholders check can now be case-insensitive as well.

Todos los cambios a detalle.

4.18 Weblate 4.12.2

Publicada el 11 de mayo de 2022.

- Se arregló la reconstrucción de la memoria de traducción a nivel proyecto para determinados componentes.
- Es posible nuevamente ordenar los componentes por cadenas sin traducir.
- Se corrigió la posible pérdida de traducciones al añadir un idioma nuevo.
- Se asegura de que la clave SSH de Weblate se genere durante las migraciones.

Todos los cambios a detalle.

4.19 Weblate 4.12.1

Publicada el 29 de abril de 2022.

- Se corrigió el título del mensaje en las solicitudes de incorporación.
- Se mejoró el tratamiento de los errores sintácticos en el formato Fluent.
- Se arregló la visualización del avatar en los mensajes de notificación por correo electrónico.
- Es posible utilizar la monetización web.
- Se reparó la eliminación de las cadenas de origen antiguas al eliminar las traducciones.

Todos los cambios a detalle.

4.20 Weblate 4.12

Publicada el 20 de abril de 2022.

- *Punto final desigual* es compatible con el amárico.
- *Signo de interrogación desigual* es compatible con el birmano.
- Opciones ampliadas para el complemento *Generación de pseudorregiones*.
- Se añadió el indicador `ignore-all-checks` para pasar por alto todas las comprobaciones de calidad en una cadena.
- Avoid *Generación de pseudorregiones* add-on to trigger failing checks.
- Se añadió compatibilidad con *Solicitudes de incorporación de Gitea*.
- Se añadieron códigos de idioma estilo Linux a *Estilo de código de idioma*.
- Added support for rebuilding project translation memory.
- Improved API for creating components from a file.
- Add copy and clone buttons to other translations.
- Make merge request message configurable at component level.
- Improved maximal length restriction behavior with XML tags.
- Fixed loading Fluent files with additional comments.

Todos los cambios en detalle.

4.21 Weblate 4.11.2

Publicada el 4 de marzo de 2022.

- Se arreglaron los daños en los archivos MO binarios.

Todos los cambios a detalle.

4.22 Weblate 4.11.1

Publicada el 4 de marzo de 2022.

- Fixed missing sanitizing of arguments to Git and Mercurial - CVE-2022-23915, see GHSA-3872-f48p-pxqj for more details.
- Fixed loading fuzzy strings from CSV files.
- Added support for creating teams using the API.
- Fixed user mention suggestions display.
- The project tokens access can now be customized.

Todos los cambios en detalle.

4.23 Weblate 4.11

Released on February 25th 2022.

- Fixes stored XSS - CVE-2022-24710, see GHSA-6jp6-9rf9-gc66 for more details.
- Se arregló la instalación de complementos a través de la API.
- Se cambió el nombre de *Cadenas que necesitan intervención* a *Cadenas sin terminar*.
- Se corrigieron falsos positivos de *Sintaxis MessageFormat de ICU*.
- Indicate lock and contributor agreement on other occurrences listing.
- Fixed updating PO files with obsolete strings or missing plurals.
- Improved squash add-on compatibility with Gerrit.
- Automatically initialize user languages based on the Accept-Language header.
- Improved error handling on string removal.
- Weblate ahora requiere Python 3.7 o posterior.
- Fixed some write operations with project token authentication.
- Se ha corregido el seguimiento del estado de las cadenas cuando éstas cambian en el repositorio.
- Realice un seguimiento de los cambios de cadena desde el repositorio.
- Sticky header on translations listing to improve navigation.
- Fixed untranslating strings in *Propiedades de Java*.
- Fixed Git operation with non-ascii branch names.
- New add-on *Prerrellenar traducción con original*.
- Added *Merge without fast-forward* *Estilo de fusión*.
- Fixed *Traducción automática* add-on trigger on newly added strings.
- Improved punctuation checks for Burmese.

- Added support for defining custom teams at project level to grant users access, see *Gestionar el control de acceso por proyecto*.
- Added documentation links to alerts.
- Docker container automatically enables TLS/SSL for outgoing e-mail when needed.
- Added support for searching for resolved comments.
- Added support for borgbackup 1.2.
- Fixed applying of *Automatically translated* label.

Todos los cambios en detalle.

4.24 Weblate 4.10.1

Publicada el 22 de diciembre de 2021.

- Se documentarion los cambios introducidos por la actualización a Django 4.0.
- Fixed displaying of *Automatically translated* label.
- Fixed API display of branch in components with a shared repository.
- Improved analysis on the failed push alert.
- Fixed manually editing page when browsing changes.
- Se mejoró la precisión de *Kashida utilizado*.
- El contenedor Docker de Weblate ahora utiliza Python 3.10.

Todos los cambios en detalle.

4.25 Weblate 4.10

Publicada el 16 de diciembre de 2021.

- Se añadió compatibilidad con niveles de formalidad y sustitutivos con DeepL.
- Bulk edit and search and replace are now available on project and language level.
- Se añadió la posibilidad de filtrar en la función Buscar y reemplazar.
- Fixed: «Perform automatic translation» privilege is no longer part of the *Languages* group.
- «Perform automatic translation» is in the *Administration* and the new *Automatic translation* group.
- Fixed generating XLSX files with special chars.
- Added ability to the GitHub authentication backend to check if the user belongs to a specific GitHub organization or team.
- Improved feedback on invalid parameters passed to API.
- Added support for project scoped access tokens for API.
- Fixed string removal in some cases.
- Fixed translating newly added strings.
- Label automatically translated strings to ease their filtering.

Todos los cambios en detalle.

4.26 Weblate 4.9.1

Publicada el 19 de noviembre de 2021.

- Fixed upload of monolingual files after changing template.
- Improved handling of whitespace in flags.
- Add support for filtering in download API.
- Se corrigieron las estadísticas que se muestran al agregar nuevas traducciones.
- Mitigate issues with GitHub SSH key change.

Todos los cambios en detalle.

4.27 Weblate 4.9

Publicada el 10 de noviembre de 2021.

- Provide more details for events in history.
- Improved rendering of history.
- Improved performance of the translation pages.
- Added support for restricting translation file downloads.
- The `safe-html` can now understand Markdown when used with `md-text`.
- The `max-length` tag now ignores XML markup when used with `xml-text`.
- Fixed dimensions of rendered texts in *Tamaño máximo de la traducción*.
- Lowered app store title length to 30 to assist with upcoming Google policy changes.
- Added support for customizing SSH invocation via *SSH_EXTRA_ARGS*.
- Se añadieron comprobaciones para el MessageFormat de ICU.
- Improved error condition handling in machine translation backends.
- Highlight unusual whitespace characters in the strings.
- Added option to stay on translated string while editing.
- Added support for customizing Borg invocation via *BORG_EXTRA_ARGS*.
- Fixed generating of MO files for monolingual translations.
- Added API endpoint to download all component translations as a ZIP file.
- Added support for Python 3.10.
- Added support for resending e-mail invitation from the management interface.

Todos los cambios en detalle.

4.28 Weblate 4.8.1

Publicada el 10 de septiembre de 2021.

- Se arregló la eliminación de usuarios en la interfaz administrativa de Django.
- Document add-on parameters in greater detail.
- Fixed JavaScript error in glossary.
- Add limit to number of matches in consistency check.
- Se mejoró la manipulación de los sustitutivos en las traducciones automáticas.
- Fixed creating add-ons using API.
- Added *PRIVACY_URL* setting to add privacy policy link to the footer.
- Hide member e-mail addresses from project admins.
- Improved gettext PO merging in case of conflicts.
- Improved glossary highlighting.
- Improved `safe-html` flag behavior with XML checks.
- Fixed commit messages for linked components.

Todos los cambios en detalle.

4.29 Weblate 4.8

Released on August 21th 2021.

- Added support for Apple stringsdict format.
- The exact search operator is now case-sensitive with PostgreSQL.
- Fixed saving glossary explanations in some cases.
- Documentation improvements.
- Performance improvements.
- Improved squash add-on compatibility with Gerrit.
- Fixed adding strings to monolingual glossary components.
- Improved performance in handling variants.
- Fixed squash add-on sometimes skipping parsing upstream changes.
- Preserve file extension for downloads.
- Added support for the Fluent format.
- Added support for using tabs to indent JSON formats.

Todos los cambios en detalle.

4.30 Weblate 4.7.2

Released on July 15th 2021.

- Support more language aliases to be configured on a project.
- Fixed search string validation in API.
- Fixed Git exporter URLs after a domain change.
- Fixed cleanup add-on for Windows RC files.
- Fixed possible crash in XLIFF updating.

Todos los cambios en detalle.

4.31 Weblate 4.7.1

Released on June 30th 2021.

- Improved popup for adding terms to glossary.
- Added support for LibreTranslate machine translation service.
- Added rate limiting on creating new projects.
- Improved performance of file updates.

Todos los cambios en detalle.

4.32 Weblate 4.7

Released on June 17th 2021.

- Improved configuration health check.
- Added support for `object-pascal-format` used in gettext PO, see *Formato de Object Pascal*.
- Cambio de nombre de *Claves cercanas* a *Claves similares* para describir mejor el propósito.
- Se añadió compatibilidad con *mi18n lang files*.
- Se mejoró la integración con la autenticación SAML.
- Fixed *Gerrit* integration to better handle corner cases.
- Weblate now requires Django 3.2.
- Fixed inviting users when e-mail authentication is disabled.
- Definiciones de lenguaje mejoradas.
- Added support for blocking users from contributing to a project.
- Fixed automatic creation of glossary languages.
- Extended documentation about add-ons.
- Performance improvements for components with linked repositories.
- Added support for free DeepL API.
- The user management no longer needs Django admin interface.

Todos los cambios en detalle.

4.33 Weblate 4.6.2

Released on May 8th 2021.

- Fixed crash after moving shared component between projects.
- Fixed adding new strings to empty properties files.
- Fixed copy icon alignment in RTL languages.
- Extended string statistics on the Info tab.
- Fixed handling of translation files ignored in Git.
- Improved metrics performance.
- Fixed possible bug in saving glossaries.
- Fixed consistency check behavior on languages with different plural rules.

Todos los cambios en detalle.

4.34 Weblate 4.6.1

Released on May 2nd 2021.

- Remove obsolete spam protection code.
- Improve source plural check accuracy.
- Update list of user interface languages in Docker.
- Improved error messages when creating pull requests.
- Fixed creating pull requests on Pagure.
- Fixed triggering automatically installed add-ons.
- Fixed possible caching issues on upgrade.
- Fixed adding new units to monolingual translations using upload.

Todos los cambios en detalle.

4.35 Weblate 4.6

Released on April 19th 2021.

- The auto_translate management command has now a parameter for specifying translation mode.
- Added support for *Archivos de texto*.
- Added trends and metrics for all objects.
- Added support for directly copying text from secondary languages.
- Added date filtering when browsing changes.
- Se remozaron los gráficos de actividad.
- Sender for contact form e-mails can now be configured.
- Improved parameters validation in component creation API.
- The rate limiting no longer applies to superusers.
- Improved automatic translation add-on performance and reliability.

- The rate limiting now can be customized in the Docker container.
- API for creating components now automatically uses *URL internos de Weblate*.
- Indicación de estado simplificada en las listas de cadenas.
- Password hashing now uses Argon2 by default.
- Barras de progreso simplificadas que indican el estado de la traducción.
- Renamed *Adición de idiomas faltantes* to clarify the purpose.
- Fixed saving string state to XLIFF.
- Se agregó búsqueda en todo el idioma.
- Initial support for *Scaling horizontally* the Docker deployment.

Todos los cambios en detalle.

4.36 Weblate 4.5.3

Publicada el 1.º de abril de 2021.

- Fixed metrics collection.
- Fixed possible crash when adding strings.
- Improved search query examples.
- Fixed possible loss of newly added strings on replace upload.

4.37 Weblate 4.5.2

Released on March 26th 2021.

- Programación configurable para traducciones automáticas.
- Added Lua format check.
- Se pasan por alto las cadenas de formato en la comprobación *Palabras consecutivas duplicadas*.
- Se permite cargar capturas de pantalla directamente desde una página de traducción.
- Se añadió la sincronización de archivos forzada en el área de mantenimiento del repositorio.
- Fixed automatic suggestions for languages with a longer code.
- Improved performance when adding new strings.
- Several bug fixes in quality checks.
- Several performance improvements.
- Integración añadida con *Descubrir Weblate*.
- Fixed checks behavior with read-only strings.

Todos los cambios en detalle.

4.38 Weblate 4.5.1

Released on March 5th 2021.

- Fixed editing of glossary flags in some corner cases.
- Extend metrics usage to improve performance of several pages.
- Se almacena el idioma de origen correcto en los archivos TMX.
- Better handling for uploads of monolingual PO using API.
- Improved alerts behavior on glossary components.
- Improved Markdown link checks.
- Indicate glossary and source language in breadcrumbs.
- Paginated component listing of huge projects.
- Improved performance of translation, component or project removal.
- Improved bulk edit performance.
- Fixed preserving «Needs editing» and «Approved» states for ODF files.
- Interfaz mejorada para personalizar las descargas de los archivos de la traducción

Todos los cambios en detalle.

4.39 Weblate 4.5

Publicada el 19 de febrero de 2021.

- Se añadió compatibilidad con `lua-format` tal como se usa en los archivos PO de gettext.
- Se agregó la posibilidad de compartir un componente entre proyectos.
- Se arregló el comportamiento de la comprobación de varias variables sin nombre en diversos indicadores de formato.
- Se desechó el campo Lista de correo en los proyectos en favor de instrucciones para traductores genéricas.
- Added pseudolocale generation add-on.
- Se pueden utilizar ahora archivos TermBase eXchange.
- Se añadió la posibilidad de definir manualmente las variantes de una cadena a través de un indicador.
- Se mejoró el desempeño de las comprobaciones de coherencia.
- Se mejoró el desempeño de la memoria de traducción en cadenas extensas.
- Se permite efectuar búsquedas en las explicaciones.
- Ahora se pueden añadir y quitar cadenas también en los formatos bilingües.
- Se amplió la lista de idiomas admitidos en la traducción automática de Amazon Translate.
- Activar automáticamente las comprobaciones de MessageFormat de Java para las .properties de Java.
- Se añadió un método de carga nuevo para añadir cadenas nuevas a una traducción.
- Se añadió una interfaz simple para explorar las traducciones.
- Los glosarios ahora se almacenan como componentes normales.
- Se eliminó la API específica de los glosarios, dado que ahora se utiliza la API de los componentes para el mismo fin.
- Se añadió una interfaz simplificada para activar o desactivar determinados indicadores.

- Se permite marcar términos como de no traducción o como prohibidos en los glosarios.
- Se añadió la capacidad de definir terminología en un glosario.
- Se movió el control de dirección del texto para obtener más espacio para el teclado virtual.
- Se añadió una opción para monitorizar automáticamente los proyectos en los que se contribuye.
- Se añadió una comprobación para asegurar que las traducciones coincidan con el glosario.
- Se permite personalizar el color de texto de la navegación.

Todos los cambios en detalle.

4.40 Weblate 4.4.2

Publicada el 14 de enero de 2021.

- Se arregló el daño en un archivo MO distribuido.

4.41 Weblate 4.4.1

Publicada el 13 de enero de 2021.

- Se corrigió la reversión de cambios con plurales.
- Se arregló la visualización de la ayuda en la configuración del proyecto.
- Se perfeccionó la administración de usuarios.
- Se manipulan mejor los contextos de los archivos PO monolingües.
- Fixed cleanup add-on behavior with HTML, ODF, IDML and Windows RC formats.
- Se arregló el procesamiento de la ubicación en los archivos CSV.
- Se utiliza compresión de contenido en las descargas de archivos.
- Se mejoró la experiencia de usuario al importar a partir de un archivo ZIP.
- Se mejoró la detección de formato de archivo durante las cargas.
- Se evita la duplicación de las solicitudes de incorporación en Pagure.
- Se mejoró el rendimiento al mostrar traducciones fantasma.
- Se reimplementó el editor de traducciones para que utilice el elemento «textarea» nativo del navegador.
- Fixed cleanup add-on breaking adding new strings.
- Added API for add-ons.

Todos los cambios en detalle.

4.42 Weblate 4.4

Publicada el 15 de diciembre de 2020.

- Se perfeccionó la validación al crear componentes.
- Weblate ahora requiere Django 3.1.
- Ahora es posible personalizar la apariencia en la interfaz de gestión.
- Fixed read-only state handling in bulk edit.

- Se mejoró la integración con CodeMirror.
- Added add-on to remove blank strings from translation files.
- Ahora se utiliza el editor CodeMirror para las traducciones.
- Coloración sintáctica en el editor de traducciones para XML, HTML, Markdown y reStructuredText.
- Resalte los elementos de ubicación en el editor de traducción.
- Se mejoró la compatibilidad con códigos de idioma no normalizados.
- Se añadió una alerta para cuando se utilizan códigos de idioma ambiguos.
- Al usuario se ofrece ahora una lista de idiomas filtrada al momento de añadir una traducción nueva.
- Se ampliaron las capacidades de búsqueda de cambios en el historial.
- Improved billing detail pages and Libre hosting workflow.
- Extended translation statistics API.
- Se mejoró la pestaña «Otras traducciones» al traducir.
- Se añadió una API para tareas.
- Improved performance of file upload.
- Se mejoró la visualización de los caracteres especiales definidos por el usuario.
- Se mejoró el desempeño de la traducción automática.
- Diversas mejoras menores en la interfaz de usuario.
- Se mejoró el nombre de las descargas ZIP.
- Se añadió una opción para recibir notificaciones de proyectos no monitorizados.

  Todos los cambios en detalle.

4.43 Weblate 4.3.2

Publicada el 4 de noviembre de 2020.

- Fixed crash on certain component file masks.
- Improved accuracy of the consecutive duplicated words check.
- Added support for Pagure pull requests.
- Se mejoraron los mensajes de error al producirse errores en el alta.
- Se revirtió la representación de los comentarios de los programadores como Markdown.
- Se simplificó la puesta en marcha de los repositorios Git con ramas predeterminadas distintas de «master».
- Los repositorios internos de nueva creación ahora utilizan «main» como rama predeterminada.
- Reduced false positives rate of unchanged translation while translating reStructuredText.
- Se arreglaron problemas de visualización de CodeMirror en algunas situaciones.
- Se cambió el nombre del grupo Plantilla a «Orígenes» para clarificar su significado.
- Fixed GitLab pull requests on repositories with longer paths.

Todos los cambios en detalle.

4.44 Weblate 4.3.1

Publicada el 21 de octubre de 2020.

- Se mejoró el desempeño de la traducción automática.
- Se arregló la caducidad de la sesión para usuarios autenticados.
- Se puede ocultar la información de versión.
- Se mejoró la compatibilidad de los actuadores con Bitbucket Server.
- Se mejoró el desempeño de las actualizaciones a la memoria de traducción.
- Se redujo el uso de memoria.
- Se mejoró el desempeño de la vista matricial.
- Se añadió una confirmación previa a la eliminación de un usuario de un proyecto.

  Todos los cambios en detalle.

4.45 Weblate 4.3

Publicada el 15 de octubre de 2020.

- La API incluye las estadísticas de cada usuario.
- Se perfeccionó la ordenación de componentes en páginas paginadas.
- Puede definirse el idioma de origen de cada glosario.
- Se reescribió la funcionalidad de creación de solicitudes de incorporación de GitHub y GitLab.
- Se corrigieron los recuentos estadísticos tras eliminar sugerencias.
- Se amplió el perfil público de usuario.
- Se corrigió la configuración de las comprobaciones obligatorias.
- Se mejoró la documentación sobre las copias de respaldo incorporadas.
- Se movió el atributo de idioma de origen del proyecto a cada componente.
- Añade la comprobación «Formato de Vue I18n».
- La comprobación de sustitutivos genéricos ahora admite expresiones regulares.
- Se mejoró el aspecto de la vista matricial.
- «Automatización» ahora se llama «Sugerencias automáticas».
- Se permite la interacción con varios ejemplares de GitLab y GitHub.
- Se amplió la API para abarcar actualizaciones de proyectos, actualizaciones y eliminaciones de unidades, así como los glosarios.
- La API de las unidades ahora maneja de manera apropiada las cadenas con formas plurales.
- Component creation can now handle ZIP file or document upload.
- Se consolidaron los códigos de estado de respuesta de la API.
- Se admite Markdown en el acuerdo de contribuidor.
- Seguimiento de cadenas de origen mejorado.
- Se perfeccionó la compatibilidad con los formatos JSON, YAML y CSV.
- Ahora es posible eliminar cadenas.
- Se mejoró el rendimiento de descarga de archivos.

- Se renovó la vista de gestión de repositorios.
- Se activa automáticamente java-format para Android.
- Se permite cargar capturas de pantalla regionalizadas.
- Se admite Python 3.9.
- Se arregló la traducción de HTML en ciertas condiciones.

Todos los cambios en detalle.

4.46 Weblate 4.2.2

Publicada el 2 de septiembre de 2020.

- Se solucionó un problema con el relacionamiento de cadenas de origen en los formatos JSON.
- Se arregló la redirección al acceder a la cuenta en determinadas configuraciones de autenticación.
- Se corrigió la autenticación LDAP con sincronización de grupo.
- Se solventó un cierre inesperado al informar del progreso automático de la traducción.
- Se arregló la concentración de consignas de Git con renglones finales activados.
- Se reparó la creación de componentes de sistema de control de versiones mediante la API.

4.47 Weblate 4.2.1

Publicada el 21 de agosto de 2020.

- Se corrigió el guardado de plurales en recursos de Android en algunas configuraciones regionales.
- Fixed crash in the cleanup add-on for some XLIFF files.
- Se permite configurar la CDN en la imagen de Docker.

4.48 Weblate 4.2

Publicada el 18 de agosto de 2020.

- Páginas de usuario mejoradas y adición de listados de usuarios.
- Se eliminó la posibilidad de migrar desde las versiones 3.x; puede migrar a través de la 4.1 o la 4.0.
- Se añadieron exportaciones a varios formatos monolingües.
- Se remozaron los gráficos de actividad.
- Es posible configurar el número de cadenas cercanas mostradas.
- Se permite bloquear los componentes cuando se producen errores en el repositorio.
- Se simplificó la navegación principal (se sustituyeron los botones por iconos).
- Se mejoró el tratamiento de los códigos de idioma en la integración con Google Translate.
- The Git squash add-on can generate `Co-authored-by:` trailers.
- Ha mejorado el procesador de consultas de búsqueda.
- Improved user feedback from format strings checks.
- Improved performance of bulk state changes.

- Added compatibility redirects after project or component renaming.
- Se añadieron notificaciones sobre aprobaciones de cadenas, bloqueos de componentes y cambios de licencia.
- Se añadió compatibilidad con ModernMT.
- Allow to avoid overwriting approved translations on file upload.
- Dropped support for some compatibility URL redirects.
- Se añadió una comprobación para los literales de plantillas ECMAScript.
- Se añadió la posibilidad de monitorizar componentes.
- Removed leading dot from JSON unit keys.
- Se eliminó la cola de Celery dedicada a la memoria de traducción.
- Se permite traducir todos los componentes en un idioma a la vez.
- Se permite configurar cabeceras `Content-Security-Policy` de HTTP.
- Added support for aliasing languages at project level.
- New add-on to help with HTML or JavaScript localization, see *CDN de regionalización de JavaScript*.
- El dominio de Weblate se establece ahora en la configuración; vea *SITE_DOMAIN*.
- Se permite buscar por componente y proyecto.

4.49 Weblate 4.1.1

Publicada el 19 de junio de 2020.

- Fixed changing autofix or add-ons configuration in Docker.
- Fixed possible crash in «About» page.
- Improved installation of byte-compiled locale files.
- Se arregló la adición de palabras al glosario.
- Se corrigieron los atajos de teclado de la automatización.
- Se quitó la salida de depuración que causaba que se descartasen sucesos del registro en algunas configuraciones.
- La vista de bloqueo en la enumeración de los proyectos ahora funciona como es debido.
- Se arregló la visualización de las claves GPG en determinados montajes.
- Se permite seleccionar qué versión de la API de DeepL utilizar.
- Added support for acting as SAML Service Provider, see *Autenticación por SAML*.

4.50 Weblate 4.1

Publicada el 15 de junio de 2020.

- Pueden crearse traducciones nuevas con código de país.
- Es posible buscar cadenas de origen con captura de pantalla.
- Información ampliada disponible en el seguimiento de estadísticas.
- Se mejoró la edición de la búsqueda en las páginas «Traducir».
- Mejoró el tratamiento de actualizaciones de repositorio concurrentes.
- Se incluye el idioma de origen en el formulario de creación de proyecto.

- Se incluye el recuento de cambios en los créditos.
- Se arregló la selección del idioma de interfaz en casos puntuales.
- Allow to whitelist registration methods with registrations closed.
- Se mejoró la búsqueda de términos relacionados en el glosario.
- Improved translation memory matches.
- Se agrupan los resultados idénticos en la automatización.
- Se añadió un enlace directo para editar capturas de pantalla desde la página de traducción.
- Se mejoró el cuadro de diálogo de confirmación ante eliminaciones.
- Se incluyen las plantillas en las descargas ZIP.
- Se permite la utilización de Markdown y la configuración de notificaciones en los anuncios.
- Detalles ampliados en los listados de comprobaciones.
- Se añadió compatibilidad con formatos de archivo nuevos: *Cadenas PHP de Laravel*, *Archivos HTML*, *Formato OpenDocument*, *Formato IDML*, *Archivos RC de Windows*, *Traducciones en INI*, *Traducciones INI de Inno Setup*, *Propiedades GWT*, *Archivos JSON de go-i18n*, *Archivo ARB*.
- Consistently use dismissed as state of dismissed checks.
- Add support for configuring default add-ons to enable.
- Se reparó el atajo de teclado en el editor para pasar por alto las comprobaciones.
- Se mejoró la traducción automática de cadenas con sustitutivos.
- Show ghost translation for user languages to ease starting them.
- Se mejoró el procesamiento de los códigos de idioma.
- Se muestran en primer lugar las traducciones en los idiomas del usuario.
- Se cambió el nombre de las «formas» a «variantes», una denominación más general.
- Se añadieron comprobaciones de calidad nuevas: *Varias variables sin nombre*, *Largamente no traducida* y *Palabras consecutivas duplicadas*.
- Se reintrodujo la posibilidad de vaciar la memoria de traducción.
- Se reparó la opción para ignorar las comprobaciones en las cadenas de origen.
- Se añadió compatibilidad para configurar una rama diferente a la que enviar cambios.
- La API ahora informa del estado de limitación de velocidad en las cabeceras HTTP.
- Se incorporó la versión 3 (avanzada) de la API de Google Translate.
- Se añadió la capacidad de restringir el acceso a nivel de componente.
- Added support for whitespace and other special chars in translation flags, see *Personalizar el comportamiento mediante indicadores*.
- Always show rendered text check if enabled.
- La API ahora admite filtrar los cambios.
- Se agregó la posibilidad de compartir glosarios entre proyectos.

4.51 Weblate 4.0.4

Released on May 7th 2020.

- Se arregló la ejecución del conjunto de pruebas en determinados entornos con Python 3.8.
- Typo fixes in the documentation.
- Fixed creating components using API in some cases.
- Fixed JavaScript errors breaking mobile navigation.
- Fixed crash on displaying some checks.
- Fixed screenshots listing.
- Fixed monthly digest notifications.
- Fixed intermediate translation behavior with units non existing in translation.

4.52 Weblate 4.0.3

Released on May 2nd 2020.

- Fixed possible crash in reports.
- Las menciones a usuarios en los comentarios ya no distinguen entre mayúsculas y minúsculas.
- Fixed PostgreSQL migration for non superusers.
- Fixed changing the repository URL while creating component.
- Se arregló un cierre inesperado cuando el repositorio de origen ascendente desaparece.

4.53 Weblate 4.0.2

Publicada el 27 de abril de 2020.

- Improved performance of translation stats.
- Improved performance of changing labels.
- Improved bulk edit performance.
- Improved translation memory performance.
- Fixed possible crash on component deletion.
- Fixed displaying of translation changes in some corner cases.
- Improved warning about too long celery queue.
- Fixed possible false positives in the consistency check.
- Fixed deadlock when changing linked component repository.
- Included edit distance in changes listing and CSV and reports.
- Avoid false positives of punctuation spacing check for Canadian French.
- Se mejoró la exportación a XLIFF con sustitutivos.
- Fixed false positive with zero width check.
- Improved reporting of configuration errors.
- Fixed bilingual source upload.

- Automatically detect supported languages for DeepL machine translation.
- Fixed progress bar display in some corner cases.
- Fixed some checks triggering on non translated strings.

4.54 Weblate 4.0.1

Publicada el 16 de abril de 2020.

- Se arregló la instalación del paquete desde PyPI.

4.55 Weblate 4.0

Publicada el 16 de abril de 2020.

- Weblate ahora requiere Python 3.6 o una versión más reciente.
- Added management overview of component alerts.
- Added component alert for broken repository browser URLs.
- Se mejoraron las páginas de alta y acceso.
- Project access control and workflow configuration integrated to project settings.
- Added check and highlighter for i18next interpolation and nesting.
- Se agregó la verificación y el resaltador para los marcadores de la posición del porcentaje.
- Se muestran las comprobaciones fallidas de las sugerencias.
- Record source string changes in history.
- Se actualizó Microsoft Translator a la versión 3 de la API.
- Se reescribió el componente dorsal de la memoria de traducción.
- Added support for several `is:` lookups in *Búsquedas*.
- Allow to make *Traducción no modificada* avoid internal blacklist.
- Improved comments extraction from monolingual po files.
- Se cambió el nombre de la función de mensajes en pizarra a «Anuncios».
- Fixed occasional problems with registration mails.
- Improved LINGUAS update add-on to handle more syntax variants.
- Fixed editing monolingual XLIFF source file.
- Added support for exact matching in *Búsquedas*.
- Extended API to cover screenshots, users, groups, componentlists and extended creating projects.
- Add support for source upload on bilingual translations.
- Added support for intermediate language from developers.
- Added support for source strings review.
- Extended download options for platform wide translation memory.

4.56 Serie 3.x de Weblate

4.56.1 Weblate 3.11.3

Publicada el 11 de marzo de 2020.

- Se corrigió la búsqueda de campos con una determinada prioridad.
- Se corrigió la consulta predefinida para cadenas añadidas recientemente.
- Se solucionó la duplicación de resultados en la búsqueda.
- Se reparó la representación gráfica de las notificaciones en Gmail.
- Se corrigió la reversión de cambios desde el historial.
- Se añadieron enlaces a sucesos en las notificaciones de resumen.
- Se corrigió la dirección de correo en la confirmación de eliminación de cuenta.
- Se permite la autenticación con Slack en contenedores Docker.
- Se evita enviar notificaciones para idiomas a los que no se ha suscrito.
- Se incluyen las colas de Celery en la vista de conjunto del rendimiento.
- Fixed documentation links for add-ons.
- Reduced false negatives for unchanged translation check.
- Se incrementó la dependencia a bleach para solucionar la vulnerabilidad CVE-2020-6802.
- Fixed listing project level changes in history.
- Fixed stats invalidation in some corner cases.
- Se solucionó la búsqueda de determinados estados de cadenas.
- Improved format string checks behavior on missing percent.
- Se reparó la autenticación a través de determinados proveedores de terceros.

4.56.2 Weblate 3.11.2

Publicada el 22 de febrero de 2020.

- Se corrigió la representación gráfica de las sugerencias.
- Se corrigió un problema por el cual algunas cadenas incorrectamente informaban que no contenían ninguna palabra.

4.56.3 Weblate 3.11.1

Publicada el 20 de febrero de 2020.

- Se documentaron modificaciones a la configuración de Celery.
- Se mejoró la validación de nombres de archivo al crear componentes.
- Se corrigieron las versiones mínimas de algunas dependencias.
- Se reparó la adición de grupos con algunas versiones de Django.
- Fixed manual pushing to upstream repository.
- Se mejoró el relacionamiento del glosario.

4.56.4 Weblate 3.11

Publicada el 17 de febrero de 2020.

- Allow using VCS push URL during component creation via API.
- Rendered width check now shows image with the render.
- Se repararon los enlaces en los mensajes de notificación por correo electrónico.
- Se mejoró el aspecto de los mensajes de correo de texto sencillo.
- Se muestran permanentemente las comprobaciones ignoradas para permitir activarlas nuevamente.
- Se muestran las claves cercanas en las traducciones monolingües.
- Se permite agrupar las formas de las cadenas.
- Se recomienda actualizar Weblate en las comprobaciones del sistema.
- Provide more detailed analysis for duplicate language alert.
- Se detalló la información sobre licencias en las páginas de proyecto.
- Automatically unshallow local copies if needed.
- Se arregló la descarga de cadenas pendientes de intervención.
- New alert to warn about using the same file mask twice.
- Improve XML placeables extraction.
- The `SINGLE_PROJECT` can now enforce redirection to chosen project.
- Se añadió una opción para marcar comentarios como resueltos.
- Added bulk editing of flags.
- Added support for labels.
- Added bulk edit add-on.
- Added option for *Forzar comprobaciones*.
- Se incrementó la validez predeterminada de los enlaces de confirmación.
- Se mejoró la integración con Matomo.
- Fixed *Se había traducido* to correctly handle source string change.
- Extended automatic updates configuration by `AUTO_UPDATE`.
- LINGUAS add-ons now do full sync of translations in Weblate.

4.56.5 Weblate 3.10.3

Publicada el 18 de abril de 2020.

- Compatibilidad con translate-toolkit 2.5.0.

4.56.6 Weblate 3.10.2

Publicada el 18 de abril de 2020.

- Se añadió un indicador de bloqueo para los proyectos.
- Se arregló un defecto en el CSS que provocaba parpadeos en determinados navegadores web.
- Se repararon las búsquedas en sistemas con configuraciones regionales distintas del inglés.
- Improved repository matching for GitHub and Bitbucket hooks.
- Se corrigió la migración de datos en algunas instalaciones con Python 2.7.
- Allow configuration of Git shallow cloning.
- Se mejoró el procesamiento en segundo plano de las notificaciones.
- Fixed broken form submission when navigating back in web browser.
- New add-on to configure YAML formatting.
- Fixed same plurals check to not fire on single plural form languages.
- Se arregló la búsqueda por expresiones regulares en algunos campos.

4.56.7 Weblate 3.10.1

Publicada el 9 de enero de 2020.

- Se amplió la API con la creación de traducciones.
- Fixed several corner cases in data migrations.
- Compatibilidad con Django 3.0.
- Se mejoró el desempeño de la limpieza de datos.
- Se permite personalizar el archivo security.txt.
- Improved breadcrumbs in changelog.
- Improved translations listing on dashboard.
- Improved HTTP responses for webhooks.
- Added support for GitLab merge requests in Docker container.

4.56.8 Weblate 3.10

Publicada el 20 de diciembre de 2019.

- Se perfeccionó la interfaz de usuario de la aplicación.
- Se añadió una comprobación de espacios dobles.
- Se arregló la creación de idiomas nuevos.
- Avoid sending auditlog notifications to deleted e-mails.
- Added support for read-only strings.
- Se permite el uso de Markdown en los comentarios.
- Allow placing translation instruction text in project info.
- Se añadió el botón «Copiar en el portapapeles» para los idiomas secundarios.
- Se mejoró la compatibilidad con Mercurial.
- Se mejoró el desempeño al recuperar repositorios Git.

- Se añadió una opción de búsqueda de cadenas por antigüedad.
- Se muestra el idioma de origen para todas las traducciones.
- Se muestra el contexto de las cadenas cercanas.
- Added support for notifications on repository operations.
- Improved translation listings.
- Extended search capabilities.
- Added support for automatic translation strings marked for editing.
- Avoid sending duplicate notifications for linked component alerts.
- Improve default merge request message.
- Better indicate string state in Zen mode.
- Added support for more languages in Yandex Translate.
- Se mejoró la apariencia de los mensajes de notificación por correo.
- Provide choice for translation license.

4.56.9 Weblate 3.9.1

Publicada el 28 de octubre de 2019.

- Remove some unneeded files from backups.
- Fixed potential crash in reports.
- Fixed cross database migration failure.
- Added support for force pushing Git repositories.
- Reduced risk of registration token invalidation.
- Fixed account removal hitting rate limiter.
- Added search based on priority.
- Fixed possible crash on adding strings to JSON file.
- Safe HTML check and fixup now honor source string markup.
- Avoid sending notifications to invited and deleted users.
- Fix SSL connection to redis in Celery in Docker container.

4.56.10 Weblate 3.9

Publicada el 15 de octubre de 2019.

- Se incluyen metadatos de Weblate en los archivos descargados.
- Se mejoró la IU de las comprobaciones fallidas.
- Indicate missing strings in format checks.
- Separate check for French punctuation spacing.
- Add support for fixing some of quality checks errors.
- Add separate permission to create new projects.
- Extend stats for char counts.
- Improve support for Java style language codes.

- Se añadió una nueva comprobación genérica para sustitutivos.
- Ahora se es compatible con los sustitutivos en archivos JSON para WebExtensions.
- Added support for flat XML format.
- Extended API with project, component and translation removal and creation.
- Added support for Gitea and Gitee webhooks.
- Added new custom regex based check.
- Allow to configure contributing to shared translation memory.
- Added ZIP download for more translation files.
- Make XLIFF standard compliant parsing of maxwidth and font.
- Added new check and fixer for safe HTML markup for translating web applications.
- Add component alert on unsupported configuration.
- Added automatic translation add-on to bootstrap translations.
- Extend automatic translation to add suggestions.
- Display add-on parameters on overview.
- Sentry is now supported through modern Sentry SDK instead of Raven.
- Changed example settings to be better fit for production environment.
- Added automated backups using BorgBackup.
- Split cleanup add-on for RESX to avoid unwanted file updates.
- Added advanced search capabilities.
- Allow users to download their own reports.
- Added localization guide to help configuring components.
- Added support for GitLab merge requests.
- Improved display of repository status.
- Perform automated translation in the background.

4.56.11 Weblate 3.8

Publicada el 15 de agosto de 2019.

- Added support for simplified creating of similar components.
- Added support for parsing translation flags from the XML based file formats.
- Log exceptions into Celery log.
- Improve performance of repository scoped add-ons.
- Se mejoró la apariencia de los mensajes de notificación por correo.
- Se arregló el comportamiento del restablecimiento de contraseñas.
- Se mejoró el rendimiento en la mayoría de las páginas de traducción.
- Fixed listing of languages not known to Weblate.
- Add support for cloning add-ons to discovered components.
- Add support for replacing file content with uploaded.
- Add support for translating non VCS based content.
- Added OpenGraph widget image to use on social networks.

- Added support for animated screenshots.
- Improved handling of monolingual XLIFF files.
- Avoid sending multiple notifications for single event.
- Add support for filtering changes.
- Extended predefined periods for reporting.
- Added webhook support for Azure Repos.
- New opt-in notifications on pending suggestions or untranslated strings.
- Add one click unsubscribe link to notification e-mails.
- Fixed false positives with Has been translated check.
- New management interface for admins.
- String priority can now be specified using flags.
- Added language management views.
- Add checks for Qt library and Ruby format strings.
- Added configuration to better fit single project installations.
- Notify about new string on source string change on monolingual translations.
- Added separate view for translation memory with search capability.

4.56.12 Weblate 3.7.1

Publicada el 28 de junio de 2019.

- Puesta al día de la documentación.
- Fixed some requirements constraints.
- Se actualizó la base de datos de idiomas.
- Actualización de las regionalizaciones.
- Diversas modificaciones a la interfaz de usuario.
- Improved handling of unsupported but discovered translation files.
- More verbosely report missing file format requirements.

4.56.13 Weblate 3.7

Publicada el 21 de junio de 2019.

- Added separate Celery queue for notifications.
- Use consistent look with application for API browsing.
- Include approved stats in the reports.
- Report progress when updating translation component.
- Allow to abort running background component update.
- Extend template language for filename manipulations.
- Use templates for editor link and repository browser URL.
- Indicate max length and current characters count when editing translation.
- Improved handling of abbreviations in unchanged translation check.

- Refreshed landing page for new contributors.
- Add support for configuring msgmerge add-on.
- Delay opening SMTP connection when sending notifications.
- Improved error logging.
- Allow custom location in MO generating add-on.
- Added add-ons to cleanup old suggestions or comments.
- Added option to enable horizontal mode in the Zen editor.
- Improved import performance with many linked components.
- Fixed examples installation in some cases.
- Improved rendering of alerts in changes.
- Added new horizontal stats widget.
- Improved format strings check on plurals.
- Added font management tool.
- New check for rendered text dimensions.
- Added support for subtitle formats.
- Include overall completion stats for languages.
- Added reporting at project and global scope.
- Improved user interface when showing translation status.
- New Weblate logo and color scheme.
- New look of bitmap badges.

4.56.14 Weblate 3.6.1

Publicada el 26 de abril de 2019.

- Improved handling of monolingual XLIFF files.
- Fixed digest notifications in some corner cases.
- Fixed add-on script error alert.
- Fixed generating MO file for monolingual PO files.
- Fixed display of uninstalled checks.
- Indicate administered projects on project listing.
- Allow update to recover from missing VCS repository.

4.56.15 Weblate 3.6

Publicada el 20 de abril de 2019.

- Add support for downloading user data.
- Add-ons are now automatically triggered upon installation.
- Improved instructions for resolving merge conflicts.
- Cleanup add-on is now compatible with app store metadata translations.
- Configurable language code syntax when adding new translations.

- Warn about using Python 2 with planned termination of support in April 2020.
- Extract special characters from the source string for visual keyboard.
- Extended contributor stats to reflect both source and target counts.
- Admins and consistency add-ons can now add translations even if disabled for users.
- Fixed description of toggle disabling `Language-Team` header manipulation.
- Notify users mentioned in comments.
- Removed file format autodetection from component setup.
- Fixed generating MO file for monolingual PO files.
- Added digest notifications.
- Added support for muting component notifications.
- Se añadieron notificaciones al haber nuevas alertas, mensajes en pizarra o componentes.
- Notifications for administered projects can now be configured.
- Improved handling of three letter language codes.

4.56.16 Weblate 3.5.1

Publicada el 10 de marzo de 2019.

- Fixed Celery systemd unit example.
- Fixed notifications from HTTP repositories with login.
- Fixed race condition in editing source string for monolingual translations.
- Include output of failed add-on execution in the logs.
- Improved validation of choices for adding new language.
- Se permite editar el formato de archivo en la configuración del componente.
- Se actualizaron las instrucciones de instalación para preferir Python 3.
- Mejoras de rendimiento y coherencia durante la carga de traducciones.
- El servicio de terminología de Microsoft se hizo compatible con las versiones actuales de Zeep.
- Actualización de las regionalizaciones.

4.56.17 Weblate 3.5

Publicada el 3 de marzo de 2019.

- Se mejoró el rendimiento de la memoria de traducción incorporada.
- Se añadió una interfaz para gestionar la memoria de traducción global.
- Improved alerting on bad component state.
- Se añadió una interfaz para gestionar los mensajes de pizarra.
- Add-on commit message now can be configured.
- Reduce number of commits when updating upstream repository.
- Fixed possible metadata loss when moving component between projects.
- Se perfeccionó la navegación en el modo zen.
- Se sumaron varias comprobaciones de calidad nuevas (relativas a Markdown y URL).

- Se añadió compatibilidad para archivos de metadatos de tiendas de aplicaciones.
- Es posible ahora activar o desactivar las integraciones con GitHub y Gerrit.
- Se añadió una comprobación de caracteres «kashida».
- Added option to squash commits based on authors.
- Se mejoró la compatibilidad con el formato de archivo XLSX.
- Compatibilidad con Tesseract 4.0.
- Billing add-on now removes projects for unpaid billings after 45 days.

4.56.18 Weblate 3.4

Publicada el 22 de enero de 2019.

- Se añadió compatibilidad para los sustitutivos de XLIFF.
- Celery ahora puede utilizar varias colas de tareas.
- Ahora es posible cambiar el nombre y mover proyectos y componentes.
- Se incluye el recuento de caracteres en los informes.
- Added guided adding of translation components with automatic detection of translation files.
- Pueden personalizarse los mensajes de consigna de fusión para Git.
- Se añadió una indicación visual de alertas de componente en la navegación.
- Se mejoró el rendimiento al cargar los archivos de traducción.
- New add-on to squash commits prior to push.
- Se mejoró la visualización de los cambios a la traducción.
- Changed default merge style to rebase and made that configurable.
- Better handle private use subtags in language code.
- Se mejoró el rendimiento de las actualizaciones de índice de texto completo.
- Extended file upload API to support more parameters.

4.56.19 Weblate 3.3

Publicada el 30 de noviembre de 2018.

- Added support for component and project removal.
- Improved performance for some monolingual translations.
- Se añadieron alertas de componente de traducción para destacar los problemas con una traducción.
- Expose XLIFF string resname as context when available.
- Added support for XLIFF states.
- Added check for non writable files in DATA_DIR.
- Improved CSV export for changes.

4.56.20 Weblate 3.2.2

Publicada el 20 de octubre de 2018.

- Remove no longer needed Babel dependency.
- Updated language definitions.
- Improve documentation for add-ons, LDAP and Celery.
- Fixed enabling new dos-eol and auto-java-messageformat flags.
- Fixed running setup.py test from PyPI package.
- Improved plurals handling.
- Fixed translation upload API failure in some corner cases.
- Fixed updating Git configuration in case it was changed manually.

4.56.21 Weblate 3.2.1

Publicada el 10 de octubre de 2018.

- Document dependency on backports.csv on Python 2.7.
- Fix running tests under root.
- Improved error handling in gitexport module.
- Fixed progress reporting for newly added languages.
- Correctly report Celery worker errors to Sentry.
- Fixed creating new translations with Qt Linguist.
- Fixed occasional fulltext index update failures.
- Improved validation when creating new components.
- Added support for cleanup of old suggestions.

4.56.22 Weblate 3.2

Publicada el 6 de octubre de 2018.

- Add install_addon management command for automated add-on installation.
- Allow more fine grained ratelimit settings.
- Added support for export and import of Excel files.
- Improve component cleanup in case of multiple component discovery add-ons.
- Rewritten Microsoft Terminology machine translation backend.
- Weblate now uses Celery to offload some processing.
- Improved search capabilities and added regular expression search.
- Added support for Youdao Zhiyun API machine translation.
- Added support for Baidu API machine translation.
- Integrated maintenance and cleanup tasks using Celery.
- Improved performance of loading translations by almost 25%.
- Removed support for merging headers on upload.
- Removed support for custom commit messages.

- Configurable editing mode (zen/full).
- Added support for error reporting to Sentry.
- Added support for automated daily update of repositories.
- Added support for creating projects and components by users.
- Built-in translation memory now automatically stores translations done.
- Users and projects can import their existing translation memories.
- Better management of related strings for screenshots.
- Added support for checking Java MessageFormat.

See 3.2 milestone on GitHub for detailed list of addressed issues.

4.56.23 Weblate 3.1.1

Publicada el 27 de julio de 2018.

- Fix testsuite failure on some setups.

4.56.24 Weblate 3.1

Publicada el 27 de julio de 2018.

- Upgrades from older version than 3.0.1 are not supported.
- Allow to override default commit messages from settings.
- Improve webhooks compatibility with self hosted environments.
- Added support for Amazon Translate.
- Compatibility with Django 2.1.
- Django system checks are now used to diagnose problems with installation.
- Removed support for soon shutdown libravatar service.
- Nuevo complemento para marcar las traducciones sin cambios como que necesitan edición.
- Add support for jumping to specific location while translating.
- Downloaded translations can now be customized.
- Improved calculation of string similarity in translation memory matches.
- Added support by signing Git commits by GnuPG.

4.56.25 Weblate 3.0.1

Publicada el 10 de junio de 2018.

- Fixed possible migration issue from 2.20.
- Actualización de las regionalizaciones.
- Removed obsolete hook examples.
- Improved caching documentation.
- Fixed displaying of admin documentation.
- Improved handling of long language names.

4.56.26 Weblate 3.0

Publicada el 1.º de junio de 2018.

- Rewritten access control.
- Several code cleanups that lead to moved and renamed modules.
- New add-on for automatic component discovery.
- The import_project management command has now slightly different parameters.
- Added basic support for Windows RC files.
- New add-on to store contributor names in PO file headers.
- The per component hook scripts are removed, use add-ons instead.
- Add support for collecting contributor agreements.
- Access control changes are now tracked in history.
- New add-on to ensure all components in a project have same translations.
- Support for more variables in commit message templates.
- Add support for providing additional textual context.

4.57 Serie 2.x de Weblate

4.57.1 Weblate 2.20

Publicada el 4 de abril de 2018.

- Improved speed of cloning subversion repositories.
- Changed repository locking to use third party library.
- Added support for downloading only strings needing action.
- Se permite efectuar búsquedas en varios idiomas a la vez.
- New add-on to configure gettext output wrapping.
- New add-on to configure JSON formatting.
- Added support for authentication in API using RFC 6750 compatible Bearer authentication.
- Added support for automatic translation using machine translation services.
- Se admite código HTML en los mensajes de la pizarra.
- Added support for mass changing state of strings.
- Translate-toolkit at least 2.3.0 is now required, older versions are no longer supported.
- Añadida memoria de traducción integrada.
- Added componentlists overview to dashboard and per component list overview pages.
- Added support for DeepL machine translation service.
- Machine translation results are now cached inside Weblate.
- Se añadió la funcionalidad de reordenar los cambios consignados.

4.57.2 Weblate 2.19.1

Publicada el 20 de febrero de 2018.

- Fixed migration issue on upgrade from 2.18.
- Improved file upload API validation.

4.57.3 Weblate 2.19

Publicada el 15 de febrero de 2018.

- Fixed imports across some file formats.
- Display human friendly browser information in audit log.
- Added TMX exporter for files.
- Various performance improvements for loading translation files.
- Added option to disable access management in Weblate in favor of Django one.
- Improved glossary lookup speed for large strings.
- Compatibility with django_auth_ldap 1.3.0.
- Configuration errors are now stored and reported persistently.
- Honor ignore flags in whitespace autofixer.
- Improved compatibility with some Subversion setups.
- Improved built-in machine translation service.
- Added support for SAP Translation Hub service.
- Added support for Microsoft Terminology service.
- Removed support for advertisement in notification e-mails.
- Improved translation progress reporting at language level.
- Improved support for different plural formulas.
- Added support for Subversion repositories not using stdlayout.
- Added add-ons to customize translation workflows.

4.57.4 Weblate 2.18

Publicada el 15 de diciembre de 2017.

- Extended contributor stats.
- Improved configuration of special characters virtual keyboard.
- Added support for DTD file format.
- Changed keyboard shortcuts to less likely collide with browser/system ones.
- Improved support for approved flag in XLIFF files.
- Added support for not wrapping long strings in gettext PO files.
- Added button to copy permalink for current translation.
- Dropped support for Django 1.10 and added support for Django 2.0.
- Removed locking of translations while translating.
- Added support for adding new strings to monolingual translations.

- Added support for translation workflows with dedicated reviewers.

4.57.5 Weblate 2.17.1

Publicada el 13 de octubre de 2017.

- Fixed running testsuite in some specific situations.
- Locales updates.

4.57.6 Weblate 2.17

Publicada el 13 de octubre de 2017.

- Weblate by default does shallow Git clones now.
- Improved performance when updating large translation files.
- Added support for blocking certain e-mails from registration.
- Users can now delete their own comments.
- Added preview step to search and replace feature.
- Client side persistence of settings in search and upload forms.
- Extended search capabilities.
- More fine grained per project ACL configuration.
- Default value of BASE_DIR has been changed.
- Added two step account removal to prevent accidental removal.
- Project access control settings is now editable.
- Added optional spam protection for suggestions using Akismet.

4.57.7 Weblate 2.16

Publicada el 11 de agosto de 2017.

- Various performance improvements.
- Added support for nested JSON format.
- Added support for WebExtension JSON format.
- Fixed git exporter authentication.
- Improved CSV import in certain situations.
- Improved look of Other translations widget.
- The max-length checks is now enforcing length of text in form.
- Make the commit_pending age configurable per component.
- Various user interface cleanups.
- Fixed component/project/site wide search for translations.

4.57.8 Weblate 2.15

Publicada el 30 de junio de 2017.

- Show more related translations in other translations.
- Add option to see translations of current string to other languages.
- Use 4 plural forms for Lithuanian by default.
- Fixed upload for monolingual files of different format.
- Improved error messages on failed authentication.
- Keep page state when removing word from glossary.
- Added direct link to edit secondary language translation.
- Added Perl format quality check.
- Added support for rejecting reused passwords.
- Extended toolbar for editing RTL languages.

4.57.9 Weblate 2.14.1

Publicada el 24 de mayo de 2017.

- Fixed possible error when paginating search results.
- Fixed migrations from older versions in some corner cases.
- Se solucionó un potencial ataque CSRF al efectuar operaciones de monitorización de proyectos.
- The password reset no longer authenticates user.
- Fixed possible CAPTCHA bypass on forgotten password.

4.57.10 Weblate 2.14

Publicada el 17 de mayo de 2017.

- Add glossary entries using AJAX.
- The logout now uses POST to avoid CSRF.
- The API key token reset now uses POST to avoid CSRF.
- Weblate sets Content-Security-Policy by default.
- The local editor URL is validated to avoid self-XSS.
- The password is now validated against common flaws by default.
- Notify users about important activity with their account such as password change.
- The CSV exports now escape potential formulas.
- Various minor improvements in security.
- The authentication attempts are now rate limited.
- Suggestion content is stored in the history.
- Store important account activity in audit log.
- Ask for password confirmation when removing account or adding new associations.
- Show time when suggestion has been made.
- There is new quality check for trailing semicolon.

- Ensure that search links can be shared.
- Included source string information and screenshots in the API.
- Allow to overwrite translations through API upload.

4.57.11 Weblate 2.13.1

Publicada el 12 de abril de 2017.

- Fixed listing of managed projects in profile.
- Fixed migration issue where some permissions were missing.
- Fixed listing of current file format in translation download.
- Return HTTP 404 when trying to access project where user lacks privileges.

4.57.12 Weblate 2.13

Publicada el 12 de abril de 2017.

- Fixed quality checks on translation templates.
- Se añadió una comprobación de calidad que se desencadena al perder traducciones.
- Se añadió una opción para ver las sugerencias pendientes de un usuario.
- Se añadió una opción para crear listas de componentes automáticamente.
- Es posible configurar el cuadro de mando que ven de manera predeterminada los usuarios no autenticados.
- Se añadió una opción para examinar 25 cadenas al azar para su revisión.
- El historial ahora incluye los cambios a las cadenas.
- Better error reporting when adding new translation.
- Added per language search within project.
- Group ACLs can now be limited to certain permissions.
- The per project ACLs are now implemented using Group ACL.
- Added more fine grained privileges control.
- Various minor UI improvements.

4.57.13 Weblate 2.12

Publicada el 3 de marzo de 2017.

- Improved admin interface for groups.
- Added support for Yandex Translate API.
- Improved speed of site wide search.
- Added project and component wide search.
- Added project and component wide search and replace.
- Improved rendering of inconsistent translations.
- Added support for opening source files in local editor.
- Added support for configuring visual keyboard with special characters.
- Improved screenshot management with OCR support for matching source strings.

- Default commit message now includes translation information and URL.
- Added support for Joomla translation format.
- Improved reliability of import across file formats.

4.57.14 Weblate 2.11

Publicada el 31 de enero de 2017.

- Include language detailed information on language page.
- Mercurial backend improvements.
- Added option to specify translation component priority.
- More consistent usage of Group ACL even with less used permissions.
- Added WL_BRANCH variable to hook scripts.
- Improved developer documentation.
- Better compatibility with various Git versions in Git exporter add-on.
- Included per project and component stats.
- Added language code mapping for better support of Microsoft Translate API.
- Moved fulltext cleanup to background job to make translation removal faster.
- Fixed displaying of plural source for languages with single plural form.
- Improved error handling in import_project.
- Various performance improvements.

4.57.15 Weblate 2.10.1

Publicada el 20 de enero de 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

4.57.16 Weblate 2.10

Publicada el 15 de diciembre de 2016.

- Added quality check to check whether plurals are translated differently.
- Fixed GitHub hooks for repositories with authentication.
- Added optional Git exporter module.
- Support for Microsoft Cognitive Services Translator API.
- Simplified project and component user interface.
- Added automatic fix to remove control characters.
- Added per language overview to project.
- Added support for CSV export.
- Added CSV download for stats.
- Added matrix view for quick overview of all translations.
- Added basic API for changes and strings.
- Added support for Apertium APy server for machine translations.

4.57.17 Weblate 2.9

Publicada el 4 de noviembre de 2016.

- Extended parameters for createadmin management command.
- Extended import_json to be able to handle with existing components.
- Se admiten los archivos YAML.
- Los propietarios de los proyectos ahora pueden configurar los componentes de traducción y los detalles del proyecto.
- Ahora se habla de «monitorizar» proyectos, no de «suscribirse» a estos.
- Es posible monitorizar proyectos directamente desde la página del proyecto.
- Added multi language status widget.
- Highlight secondary language if not showing source.
- Se registra la eliminación de sugerencias en el historial.
- Improved UX of languages selection in profile.
- Se arregló la visualización de los mensajes de pizarra en los componentes.
- Keep preferences tab selected after saving.
- Show source string comment more prominently.
- Automatically install Gettext PO merge driver for Git repositories.
- Added search and replace feature.
- Added support for uploading visual context (screenshots) for translations.

4.57.18 Weblate 2.8

Publicada el 31 de agosto de 2016.

- Documentation improvements.
- Translations.
- Updated bundled JavaScript libraries.
- Added list_translators management command.
- Django 1.8 is no longer supported.
- Fixed compatibility with Django 1.10.
- Added Subversion support.
- Separated XML validity check from XML mismatched tags.
- Fixed API to honor HIDE_REPO_CREDENTIALS settings.
- Show source change in Zen mode.
- Alt+PageUp/PageDown/Home/End now works in Zen mode as well.
- Add tooltip showing exact time of changes.
- Add option to select filters and search from translation page.
- Added UI for translation removal.
- Improved behavior when inserting placeables.
- Fixed auto locking issues in Zen mode.

4.57.19 Weblate 2.7

Publicada el 10 de julio de 2016.

- Removed Google web translate machine translation.
- Improved commit message when adding translation.
- Fixed Google Translate API for Hebrew language.
- Compatibility with Mercurial 3.8.
- Added import_json management command.
- Correct ordering of listed translations.
- Show full suggestion text, not only a diff.
- Extend API (detailed repository status, statistics, …).
- Testsuite no longer requires network access to test repositories.

4.57.20 Weblate 2.6

Publicada el 28 de abril de 2016.

- Fixed validation of components with language filter.
- Improved support for XLIFF files.
- Fixed machine translation for non English sources.
- Added REST API.
- Django 1.10 compatibility.
- Se añadieron categorías a los mensajes de pizarra.

4.57.21 Weblate 2.5

Publicada el 10 de marzo de 2016.

- Fixed automatic translation for project owners.
- Improved performance of commit and push operations.
- New management command to add suggestions from command-line.
- Added support for merging comments on file upload.
- Added support for some GNU extensions to C printf format.
- Documentation improvements.
- Added support for generating translator credits.
- Added support for generating contributor stats.
- Site wide search can search only in one language.
- Improve quality checks for Armenian.
- Support for starting translation components without existing translations.
- Support for adding new translations in Qt TS.
- Improved support for translating PHP files.
- Mejoras de rendimiento para las comprobaciones de calidad.
- Se arregló la búsqueda en todo el sitio de comprobaciones fallidas.

- Se añadió una opción para especificar el idioma de origen.
- Improved support for XLIFF files.
- Se amplió la lista de opciones disponibles para import_project.
- Improved targeting for whiteboard messages.
- Es posible efectuar traducciones automáticas en diversos proyectos a la vez.
- Se optimizó la indización de la búsqueda de texto completo.
- Se añadió una orden de gestión para traducciones automáticas.
- Added placeables highlighting.
- Added keyboard shortcuts for placeables, checks and machine translations.
- Improved translation locking.
- Added quality check for AngularJS interpolation.
- Added extensive group based ACLs.
- Clarified terminology on strings needing edit (formerly fuzzy).
- Clarified terminology on strings needing action and untranslated strings.
- Se admite Python 3.
- Ya no se admite Django 1.7.
- Dropped dependency on msginit for creating new gettext PO files.
- Added configurable dashboard views.
- Improved notifications on parse errors.
- Added option to import components with duplicate name to import_project.
- Improved support for translating PHP files.
- Added XLIFF export for dictionary.
- Added XLIFF and gettext PO export for all translations.
- Documentation improvements.
- Added support for configurable automatic group assignments.
- Improved adding of new translations.

4.57.22 Weblate 2.4

Publicada el 20 de septiembre de 2015.

- Improved support for PHP files.
- Ability to add ACL to anonymous user.
- Improved configurability of import_project command.
- Added CSV dump of history.
- Avoid copy/paste errors with whitespace characters.
- Added support for Bitbucket webhooks.
- Tighter control on fuzzy strings on translation upload.
- Several URLs have changed, you might have to update your bookmarks.
- Hook scripts are executed with VCS root as current directory.
- Hook scripts are executed with environment variables describing current component.

- Add management command to optimize fulltext index.
- Added support for error reporting to Rollbar.
- Projects now can have multiple owners.
- Project owners can manage themselves.
- Added support for `javascript-format` used in gettext PO.
- Support for adding new translations in XLIFF.
- Improved file format autodetection.
- Extended keyboard shortcuts.
- Improved dictionary matching for several languages.
- Improved layout of most of pages.
- Support for adding words to dictionary while translating.
- Added support for filtering languages to be managed by Weblate.
- Added support for translating and importing CSV files.
- Rewritten handling of static files.
- Direct login/registration links to third-party service if that's the only one.
- Commit pending changes on account removal.
- Add management command to change site name.
- Add option to configure default committer.
- Add hook after adding new translation.
- Add option to specify multiple files to add to commit.

4.57.23 Weblate 2.3

Publicada el 22 de mayo de 2015.

- Dropped support for Django 1.6 and South migrations.
- Support for adding new translations when using Java Property files.
- Allow to accept suggestion without editing.
- Improved support for Google OAuth 2.0.
- Added support for Microsoft .resx files.
- Tuned default robots.txt to disallow big crawling of translations.
- Simplified workflow for accepting suggestions.
- Added project owners who always receive important notifications.
- Allow to disable editing of monolingual template.
- More detailed repository status view.
- Direct link for editing template when changing translation.
- Allow to add more permissions to project owners.
- Allow to show secondary language in Zen mode.
- Support for hiding source string in favor of secondary language.

4.57.24 Weblate 2.2

Publicada el 19 de febrero de 2015.

- Performance improvements.
- Fulltext search on location and comments fields.
- New SVG/JavaScript-based activity charts.
- Support for Django 1.8.
- Support for deleting comments.
- Added own SVG badge.
- Added support for Google Analytics.
- Improved handling of translation filenames.
- Added support for monolingual JSON translations.
- Record component locking in a history.
- Support for editing source (template) language for monolingual translations.
- Added basic support for Gerrit.

4.57.25 Weblate 2.1

Publicada el 5 de diciembre de 2014.

- Added support for Mercurial repositories.
- Replaced Glyphicon font by Awesome.
- Added icons for social authentication services.
- Better consistency of button colors and icons.
- Documentation improvements.
- Varias correcciones de defectos.
- Automatic hiding of columns in translation listing for small screens.
- Changed configuration of filesystem paths.
- Improved SSH keys handling and storage.
- Improved repository locking.
- Customizable quality checks per source string.
- Allow to hide completed translations from dashboard.

4.57.26 Weblate 2.0

Publicada el 6 de noviembre de 2014.

- New responsive UI using Bootstrap.
- Rewritten VCS backend.
- Documentation improvements.
- Added whiteboard for site wide messages.
- Configurable strings priority.
- Added support for JSON file format.

- Fixed generating mo files in certain cases.
- Added support for GitLab notifications.
- Added support for disabling translation suggestions.
- Django 1.7 support.
- ACL projects now have user management.
- Extended search possibilities.
- Give more hints to translators about plurals.
- Fixed Git repository locking.
- Compatibility with older Git versions.
- Improved ACL support.
- Added buttons for per language quotes and other special characters.
- Support for exporting stats as JSONP.

4.58 Serie 1.x de Weblate

4.58.1 Weblate 1.9

Publicada el 6 de mayo de 2014.

- Django 1.6 compatibility.
- No longer maintained compatibility with Django 1.4.
- Management commands for locking/unlocking translations.
- Improved support for Qt TS files.
- Users can now delete their account.
- Avatars can be disabled.
- Merged first and last name attributes.
- Avatars are now fetched and cached server side.
- Added support for shields.io badge.

4.58.2 Weblate 1.8

Publicada el 7 de noviembre de 2013.

- Please check manual for upgrade instructions.
- Nicer listing of project summary.
- Better visible options for sharing.
- More control over anonymous users privileges.
- Supports login using third party services, check manual for more details.
- Los usuarios pueden acceder proporcionando su correo electrónico en lugar de su nombre de usuario.
- Documentation improvements.
- Improved source strings review.
- Searching across all strings.

- Better tracking of source strings.
- Captcha protection for registration.

4.58.3 Weblate 1.7

Publicada el 7 de octubre de 2013.

- Please check manual for upgrade instructions.
- Support for checking Python brace format string.
- Per component customization of quality checks.
- Detailed per translation stats.
- Changed way of linking suggestions, checks and comments to strings.
- Users can now add text to commit message.
- Support for subscribing on new language requests.
- Support for adding new translations.
- Widgets and charts are now rendered using Pillow instead of Pango + Cairo.
- Add status badge widget.
- Dropped invalid text direction check.
- Los cambios realizados a los diccionarios ahora figuran en el historial.
- Performance improvements for translation view.

4.58.4 Weblate 1.6

Publicada el 25 de julio de 2013.

- Nicer error handling on registration.
- Browsing of changes.
- Fixed sorting of machine translation suggestions.
- Improved support for MyMemory machine translation.
- Added support for Amagama machine translation.
- Various optimizations on frequently used pages.
- Highlights searched phrase in search results.
- Support for automatic fixups while saving the message.
- Tracking of translation history and option to revert it.
- Added support for Google Translate API.
- Added support for managing SSH host keys.
- Various form validation improvements.
- Various quality checks improvements.
- Performance improvements for import.
- Added support for voting on suggestions.
- Cleanup of admin interface.

4.58.5 Weblate 1.5

Publicada el 16 de abril de 2013.

- Please check manual for upgrade instructions.
- Added public user pages.
- Better naming of plural forms.
- Added support for TBX export of glossary.
- Added support for Bitbucket notifications.
- Activity charts are now available for each translation, language or user.
- Extended options of import_project admin command.
- Compatible with Django 1.5.
- Avatars are now shown using libravatar.
- Added possibility to pretty print JSON export.
- Various performance improvements.
- Indicate failing checks or fuzzy strings in progress bars for projects or languages as well.
- Added support for custom pre-commit hooks and committing additional files.
- Rewritten search for better performance and user experience.
- New interface for machine translations.
- Added support for monolingual po files.
- Extend amount of cached metadata to improve speed of various searches.
- Now shows word counts as well.

4.58.6 Weblate 1.4

Publicada el 23 de enero de 2013.

- Fixed deleting of checks/comments on string deletion.
- Added option to disable automatic propagation of translations.
- Added option to subscribe for merge failures.
- Correctly import on projects which needs custom ttkit loader.
- Added sitemaps to allow easier access by crawlers.
- Provide direct links to string in notification e-mails or feeds.
- Various improvements to admin interface.
- Provide hints for production setup in admin interface.
- Added per language widgets and engage page.
- Improved translation locking handling.
- Show code snippets for widgets in more variants.
- Indicate failing checks or fuzzy strings in progress bars.
- More options for formatting commit message.
- Fixed error handling with machine translation services.
- Improved automatic translation locking behaviour.

- Support for showing changes from previous source string.
- Added support for substring search.
- Various quality checks improvements.
- Support for per project ACL.
- Basic code coverage by unit tests.

4.58.7 Weblate 1.3

Publicada el 16 de noviembre de 2012.

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Improved quality checks processing.
- Added new checks (BBCode, XML markup and newlines).
- Support for optional rebasing instead of merge.
- Possibility to relocate Weblate (for example to run it under /weblate path).
- Support for manually choosing file type in case autodetection fails.
- Better support for Android resources.
- Support for generating SSH key from web interface.
- More visible data exports.
- New buttons to enter some special characters.
- Support for exporting dictionary.
- Support for locking down whole Weblate installation.
- Checks for source strings and support for source strings review.
- Support for user comments for both translations and source strings.
- Better changes log tracking.
- Changes can now be monitored using RSS.
- Improved support for RTL languages.

4.58.8 Weblate 1.2

Publicada el 14 de agosto de 2012.

- Weblate ahora se sirve de South para la migración de la base de datos; eche un vistazo a las instrucciones de actualización si está por actualizar.
- Fixed minor issues with linked git repos.
- New introduction page for engaging people with translating using Weblate.
- Added widgets which can be used for promoting translation projects.
- Added option to reset repository to origin (for privileged users).
- Project or component can now be locked for translations.
- Possibility to disable some translations.
- Configurable options for adding new translations.
- Configuration of git commits per project.

- Simple antispam protection.
- Better layout of main page.
- Support for automatically pushing changes on every commit.
- Support for e-mail notifications of translators.
- List only used languages in preferences.
- Improved handling of not known languages when importing project.
- Support for locking translation by translator.
- Optionally maintain `Language-Team` header in po file.
- Include some statistics in about page.
- Supports (and requires) django-registration 0.8.
- Almacenamiento en la caché de recuentos de cadenas con comprobaciones fallidas.
- Checking of requirements during setup.
- Documentation improvements.

4.58.9 Weblate 1.1

Publicada el 4 de julio de 2012.

- Improved several translations.
- Better validation while creating component.
- Added support for shared git repositories across components.
- Do not necessary commit on every attempt to pull remote repo.
- Added support for offloading indexing.

4.58.10 Weblate 1.0

Publicada el 10 de mayo de 2012.

- Improved validation while adding/saving component.
- Experimental support for Android component files (needs patched ttkit).
- Las actualizaciones de los actuadores se ejecutan en segundo plano.
- Se mejoraron las instrucciones de instalación.
- Se mejoró la navegación en el área de diccionarios.

4.59 Serie 0.x de Weblate

4.59.1 Weblate 0.9

Publicada el 18 de abril de 2012.

- Se arregló la importación de idiomas desconocidos.
- Improved listing of nearby messages.
- Se mejoraron varias comprobaciones.
- Puesta al día de la documentación.

- Added definition for several more languages.
- Various code cleanups.
- Documentation improvements.
- Changed file layout.
- Update helper scripts to Django 1.4.
- Improved navigation while translating.
- Better handling of po file renames.
- Better validation while creating component.
- Integrated full setup into syncdb.
- Added list of recent changes to all translation pages.
- Check for untranslated strings ignores format string only messages.

4.59.2 Weblate 0.8

Publicada el 3 de abril de 2012.

- Replaced own full text search with Whoosh.
- Various fixes and improvements to checks.
- New command updatechecks.
- Muchas traducciones actualizadas.
- Added dictionary for storing most frequently used terms.
- Added /admin/report/ for overview of repositories status.
- Machine translation services no longer block page loading.
- Management interface now contains also useful actions to update data.
- Records log of changes made by users.
- Ability to postpone commit to Git to generate less commits from single user.
- Possibility to browse failing checks.
- Automatic translation using already translated strings.
- New about page showing used versions.
- Django 1.4 compatibility.
- Ability to push changes to remote repo from web interface.
- Added review of translations done by others.

4.59.3 Weblate 0.7

Publicada el 16 de febrero de 2012.

- Direct support for GitHub notifications.
- Added support for cleaning up orphaned checks and translations.
- Displays nearby strings while translating.
- Displays similar strings while translating.
- Improved searching for string.

4.59.4 Weblate 0.6

Publicada el 14 de febrero de 2012.

- Added various checks for translated messages.
- Tunable access control.
- Improved handling of translations with new lines.
- Added client side sorting of tables.
- Please check upgrading instructions in case you are upgrading.

4.59.5 Weblate 0.5

Publicada el 12 de febrero de 2012.

- **Support for machine translation using following online services:**
  - Apertium
  - Microsoft Translator
  - MyMemory
- Several new translations.
- Improved merging of upstream changes.
- Better handle concurrent git pull and translation.
- Propagating works for fuzzy changes as well.
- Propagating works also for file upload.
- Fixed file downloads while using FastCGI (and possibly others).

4.59.6 Weblate 0.4

Publicada el 8 de febrero de 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

4.59.7 Weblate 0.3

Publicada el 8 de febrero de 2012.

- Better display of source for plural translations.
- New documentation in Sphinx format.
- Displays secondary languages while translating.
- Improved error page to give list of existing projects.
- New per language stats.

4.59.8 Weblate 0.2

Publicada el 7 de febrero de 2012.

- Mejor validación de determinados formularios.
- Warn users on profile upgrade.
- Remember URL for login.
- Naming of text areas while entering plural forms.
- Ampliación automática del área de traducción.

4.59.9 Weblate 0.1

Publicada el 6 de febrero de 2012.

- Versión inicial.

Índice de Módulos Python

w

wlc, 171
wlc.config, 171
wlc.main, 172

Tabla de enrutamiento HTTP

/

ANY /, 117

/api

GET /api/, 120

/api/(str:object)

GET /api/(str:object)/statistics/, 160

/api/addons

GET /api/addons/, 157
GET /api/addons/(int:id)/, 157
PUT /api/addons/(int:id)/, 157
DELETE /api/addons/(int:id)/, 157
PATCH /api/addons/(int:id)/, 157

/api/changes

GET /api/changes/, 154
GET /api/changes/(int:id)/, 154

/api/component-lists

GET /api/component-lists/, 158
GET /api/component-lists/(str:slug)/, 158
POST /api/component-lists/(str:slug)/components/, 159
PUT /api/component-lists/(str:slug)/, 158
DELETE /api/component-lists/(str:slug)/, 158
DELETE /api/component-lists/(str:slug)/components/(str:component_slug), 159
PATCH /api/component-lists/(str:slug)/, 158

/api/components

GET /api/components/, 136
GET /api/components/(string:project)/(string:component)/, 136
GET /api/components/(string:project)/(string:component)/changes/, 140
GET /api/components/(string:project)/(string:compo 141
GET /api/components/(string:project)/(string:compo 145
GET /api/components/(string:project)/(string:compo 141
GET /api/components/(string:project)/(string:compo 143
GET /api/components/(string:project)/(string:compo 143
GET /api/components/(string:project)/(string:compo 142
GET /api/components/(string:project)/(string:compo 141
GET /api/components/(string:project)/(string:compo 145
GET /api/components/(string:project)/(string:compo 143
POST /api/components/(string:project)/(string:comp 157
POST /api/components/(string:project)/(string:comp 145
POST /api/components/(string:project)/(string:comp 141
POST /api/components/(string:project)/(string:comp 142
POST /api/components/(string:project)/(string:comp 144
PUT /api/components/(string:project)/(string:compo 140
DELETE /api/components/(string:project)/(string:co 140
DELETE /api/components/(string:project)/(string:co 146
PATCH /api/components/(string:project)/(string:com 138

/api/groups

GET /api/groups/, 124
GET /api/groups/(int:id)/, 124
POST /api/groups/, 124
POST /api/groups/(int:id)/componentlists/, 127

POST /api/groups/(int:id)/components/, 126
POST /api/groups/(int:id)/languages/, 127
POST /api/groups/(int:id)/projects/, 126
POST /api/groups/(int:id)/roles/, 126
PUT /api/groups/(int:id)/, 125
DELETE /api/groups/(int:id)/, 126
DELETE /api/groups/(int:id)/componentlists/(int:component_list_id), 127
DELETE /api/groups/(int:id)/components/(int:component_id), 126
DELETE /api/groups/(int:id)/languages/(int:language_id), 127
DELETE /api/groups/(int:id)/projects/(int:project_id), 126
PATCH /api/groups/(int:id)/, 126

/api/languages

GET /api/languages/, 128
GET /api/languages/(string:language)/, 129
GET /api/languages/(string:language)/statistics/, 130
POST /api/languages/, 128
PUT /api/languages/(string:language)/, 129
DELETE /api/languages/(string:language)/, 130
PATCH /api/languages/(string:language)/, 129

/api/memory

GET /api/memory/, 151
DELETE /api/memory/(int:memory_object_id)/, 151

/api/metrics

GET /api/metrics/, 161

/api/projects

GET /api/projects/, 130
GET /api/projects/(string:project)/, 130
GET /api/projects/(string:project)/changes/, 131
GET /api/projects/(string:project)/components/, 133
GET /api/projects/(string:project)/languages/, 135
GET /api/projects/(string:project)/repository/, 132
GET /api/projects/(string:project)/statistics/, 136
POST /api/projects/, 130
POST /api/projects/(string:project)/components/, 133
POST /api/projects/(string:project)/repository/, 132
PUT /api/projects/(string:project)/, 131
DELETE /api/projects/(string:project)/, 131
PATCH /api/projects/(string:project)/, 131

/api/roles

GET /api/roles/, 127
GET /api/roles/(int:id)/, 127
POST /api/roles/, 127
PUT /api/roles/(int:id)/, 128
DELETE /api/roles/(int:id)/, 128
PATCH /api/roles/(int:id)/, 128

/api/screenshots

GET /api/screenshots/, 154
GET /api/screenshots/(int:id)/, 154
GET /api/screenshots/(int:id)/file/, 155
POST /api/screenshots/, 155
POST /api/screenshots/(int:id)/file/, 155
POST /api/screenshots/(int:id)/units/, 155
PUT /api/screenshots/(int:id)/, 156
DELETE /api/screenshots/(int:id)/, 156
DELETE /api/screenshots/(int:id)/units/(int:unit_
PATCH /api/screenshots/(int:id)/, 156

/api/search

GET /api/search/, 161

/api/tasks

GET /api/tasks/, 159
GET /api/tasks/(str:uuid)/, 159

/api/translations

GET /api/translations/, 146
GET /api/translations/(string:project)/(string:co
146
GET /api/translations/(string:project)/(string:co
148
GET /api/translations/(string:project)/(string:co
150
GET /api/translations/(string:project)/(string:co
150
GET /api/translations/(string:project)/(string:co
151
GET /api/translations/(string:project)/(string:co
149
POST /api/translations/(string:project)/(string:c
149
POST /api/translations/(string:project)/(string:c
150

POST /api/translations/(string:project)/(string:component)/(string:language)/repository/, 151
POST /api/translations/(string:project)/(string:component)/(string:language)/units/, 149
DELETE /api/translations/(string:project)/(string:component)/(string:language)/, 148
POST /hooks/bitbucket/, 162
POST /hooks/gitea/, 163
POST /hooks/github/, 162
POST /hooks/gitlab/, 162
POST /hooks/pagure/, 162

/api/units

GET /api/units/, 152
GET /api/units/(int:id)/, 152
PUT /api/units/(int:id)/, 153
DELETE /api/units/(int:id)/, 153
PATCH /api/units/(int:id)/, 153

/api/users

GET /api/users/, 121
GET /api/users/(str:username)/, 121
GET /api/users/(str:username)/notifications/, 123
GET /api/users/(str:username)/notifications/(int:subscription_id)/, 123
GET /api/users/(str:username)/statistics/, 123
POST /api/users/, 121
POST /api/users/(str:username)/groups/, 122
POST /api/users/(str:username)/notifications/, 123
PUT /api/users/(str:username)/, 122
PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 123
DELETE /api/users/(str:username)/, 122
DELETE /api/users/(str:username)/groups/, 123
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 124
PATCH /api/users/(str:username)/, 122
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 124

/exports

GET /exports/rss/, 165
GET /exports/rss/(string:project)/, 165
GET /exports/rss/(string:project)/(string:component)/, 165
GET /exports/rss/(string:project)/(string:component)/(string:language)/, 165
GET /exports/rss/language/(string:language)/, 165
GET /exports/stats/(string:project)/(string:component)/, 164

/hooks

GET /hooks/update/(string:project)/, 161
GET /hooks/update/(string:project)/(string:component)/, 161
POST /hooks/azure/, 163

Índice

Símbolos

.XML resource file
    file format, 99
--add
    opción de línea de comando auto_translate, 406
--addon
    opción de línea de comando install_addon, 412
--age
    opción de línea de comando commit_pending, 407
--author
    opción de línea de comando add_suggestions, 406
--author-email
    opción de línea de comando wlc, 169
--author-name
    opción de línea de comando wlc, 169
--base-file-template
    opción de línea de comando import_project, 410
--check
    opción de línea de comando importusers, 412
--config
    opción de línea de comando wlc, 167
--config-section
    opción de línea de comando wlc, 167
--configuration
    opción de línea de comando install_addon, 412
    opción de línea de comando install_machinery, 413
--convert
    opción de línea de comando wlc, 168
--email
    opción de línea de comando createadmin, 408
--file-format
    opción de línea de comando import_project, 410
--force
    opción de línea de comando loadpo, 414
--force-commit
    opción de línea de comando pushgit, 415
--format
    opción de línea de comando wlc, 167
--fuzzy
    opción de línea de comando wlc, 168
--ignore
    opción de línea de comando import_json, 409
--inconsistent
    opción de línea de comando auto_translate, 406
--input
    opción de línea de comando wlc, 168
--key
    opción de línea de comando wlc, 167
--lang
    opción de línea de comando loadpo, 414
--language-code
    opción de línea de comando list_translators, 413
--language-map
    opción de línea de comando import_memory, 410
--language-regex
    opción de línea de comando import_project, 410
--license
    opción de línea de comando import_project, 411
--license-url
    opción de línea de comando import_project, 411
--main-component
    opción de línea de comando import_json, 409
    opción de línea de comando import_project, 410
--method
    opción de línea de comando wlc, 168

--mode
    opción de línea de comando auto_translate, 406
--mt
    opción de línea de comando auto_translate, 406
--name
    opción de línea de comando createadmin, 408
--name-template
    opción de línea de comando import_project, 410
--new-base-template
    opción de línea de comando import_project, 410
--no-password
    opción de línea de comando createadmin, 408
--no-privs-update
    opción de línea de comando setupgroups, 416
--no-projects-update
    opción de línea de comando setupgroups, 416
--no-update
    opción de línea de comando setuplang, 416
--output
    opción de línea de comando wlc, 168
--overwrite
    opción de línea de comando auto_translate, 406
    opción de línea de comando wlc, 168
--password
    opción de línea de comando createadmin, 408
--project
    opción de línea de comando import_json, 409
--service
    opción de línea de comando install_machinery, 413
--source
    opción de línea de comando auto_translate, 406
--threshold
    opción de línea de comando auto_translate, 406
--update
    opción de línea de comando createadmin, 408
    opción de línea de comando import_json, 409
    opción de línea de comando install_addon, 412
    opción de línea de comando install_machinery, 413
--url
    opción de línea de comando wlc, 167
--user
    opción de línea de comando auto_translate, 406
--username
    opción de línea de comando createadmin, 408
--vcs
    opción de línea de comando import_project, 411

A

add_suggestions
    weblate admin command, 406
ADMINS
    setting, 223
AKISMET_API_KEY
    setting, 360
ALLOWED_HOSTS
    setting, 223
Android
    file format, 82
ANONYMOUS_USER_NAME
    setting, 360
API, 117, 165, 170
Apple strings
    file format, 83
ARB
    file format, 84
AUDITLOG_EXPIRY
    setting, 360
AUTH_LOCK_ATTEMPTS
    setting, 360
AUTH_TOKEN_VALID
    setting, 361
auto_translate
    weblate admin command, 406
AUTO_UPDATE
    setting, 361
AUTOFIX_LIST
    setting, 362
AVATAR_URL_PREFIX
    setting, 361

B

BACKGROUND_TASKS
    setting, 362
BaseAddon (*clase en weblate.addons.base*), 451
BASIC_LANGUAGES
    setting, 363
bilingual
    translation, 108
BITBUCKETSERVER_CREDENTIALS
    setting, 372
BORG_EXTRA_ARGS
    setting, 363

C

CACHE_DIR
    setting, 363
can_install() (*método de clase de weblate.addons.base.BaseAddon*), 451
CELERY_BACKUP_OPTIONS, 178, 196
CELERY_BEAT_OPTIONS, 178, 196
CELERY_MAIN_OPTIONS, 178, 196
CELERY_MEMORY_OPTIONS, 178, 196
CELERY_NOTIFY_OPTIONS, 178, 196
celery_queues
    weblate admin command, 407
CELERY_TRANSLATE_OPTIONS, 178, 196
changes
    opción de línea de comando wlc, 168
CHECK_LIST
    setting, 364
checkgit
    weblate admin command, 407
cleanup
    opción de línea de comando wlc, 168
cleanup_ssh_keys
    weblate admin command, 408
cleanuptrans
    weblate admin command, 407
Comma separated values
    file format, 85
Command (*clase en wlc.main*), 172
COMMENT_CLEANUP_DAYS
    setting, 364
commit
    opción de línea de comando wlc, 167
commit_pending
    weblate admin command, 407
COMMIT_PENDING_HOURS
    setting, 365
commitgit
    weblate admin command, 407
configure() (*método de weblate.addons.base.BaseAddon*), 451
CONTACT_FORM
    setting, 365
createadmin
    weblate admin command, 408
CSP_CONNECT_SRC
    setting, 364
CSP_FONT_SRC
    setting, 364
CSP_IMG_SRC
    setting, 364
CSP_SCRIPT_SRC
    setting, 364
CSP_STYLE_SRC
    setting, 364
CSV
    file format, 85

D

daily() (*método de weblate.addons.base.BaseAddon*), 451
DATA_DIR
    setting, 365
DATABASE_BACKUP
    setting, 366
DATABASES
    setting, 224
DEBUG
    setting, 224
DEFAULT_ACCESS_CONTROL
    setting, 366
DEFAULT_ADD_MESSAGE
    setting, 367
DEFAULT_ADDON_MESSAGE
    setting, 367
DEFAULT_ADDONS
    setting, 367
DEFAULT_AUTO_WATCH
    setting, 366
DEFAULT_COMMIT_MESSAGE
    setting, 367
DEFAULT_COMMITER_EMAIL
    setting, 367
DEFAULT_COMMITER_NAME
    setting, 368
DEFAULT_DELETE_MESSAGE
    setting, 367
DEFAULT_FROM_EMAIL
    setting, 224
DEFAULT_LANGUAGE
    setting, 368
DEFAULT_MERGE_MESSAGE
    setting, 367
DEFAULT_MERGE_STYLE
    setting, 368
DEFAULT_PAGE_LIMIT
    setting, 378
DEFAULT_PULL_MESSAGE
    setting, 368
DEFAULT_RESTRICTED_COMPONENT
    setting, 367
DEFAULT_SHARED_TM
    setting, 368
DEFAULT_TRANSLATION_PROPAGATION
    setting, 368
download
    opción de línea de comando wlc, 168
DTD
    file format, 86
dump_memory
    weblate admin command, 408
dumpuserdata
    weblate admin command, 408

E

ENABLE_AVATARS

setting, 369
ENABLE_HOOKS
setting, 369
ENABLE_HTTPS
setting, 369
ENABLE_SHARING
setting, 369
EXTRA_HTML_HEAD
setting, 369

F

file format
.XML resource file, 99
Android, 82
Apple strings, 83
ARB, 84
Comma separated values, 85
CSV, 85
DTD, 86
gettext, 88
go-i18n, 89
gotext, 90
GWT properties, 90
i18next, 91
INI translations, 92, 93
Java properties, 93
Joomla translations, 94
JSON, 94
mi18n lang, 96
PHP strings, 97
PO, 88
Qt, 98
RC, 102
ResourceDictionary, 98
RESX, 99
Ruby YAML, 99
Ruby YAML Ain't Markup Language, 99
string resources, 82
TS, 98
WPF, 98
XLIFF, 103
XML, 87
YAML, 105
YAML Ain't Markup Language, 105

G

get() (*método de wlc.Weblate*), 171
get_add_form() (*método de clase de weblate.addons.base.BaseAddon*), 451
GET_HELP_URL
setting, 369
get_settings_form() (*método de weblate.addons.base.BaseAddon*), 451
gettext
file format, 88
GITEA_CREDENTIALS
setting, 370
GITHUB_CREDENTIALS
setting, 371
GITLAB_CREDENTIALS
setting, 370
go-i18n
file format, 89
GOOGLE_ANALYTICS_ID
setting, 373
gotext
file format, 90
GWT properties
file format, 90

H

HIDE_REPO_CREDENTIALS
setting, 373
HIDE_VERSION
setting, 373

I

i18next
file format, 91
import_demo
weblate admin command, 409
import_json
weblate admin command, 409
import_memory
weblate admin command, 410
import_project
weblate admin command, 410
importuserdata
weblate admin command, 412
importusers
weblate admin command, 412
INI translations
file format, 92, 93
install_addon
weblate admin command, 412
install_machinery
weblate admin command, 413
INTERLEDGER_PAYMENT_POINTERS
setting, 373
iOS
translation, 83
IP_BEHIND_REVERSE_PROXY
setting, 374
IP_PROXY_HEADER
setting, 374
IP_PROXY_OFFSET
setting, 374

J

Java properties
file format, 93
Joomla translations
file format, 94
JSON
file format, 94

L

LEGAL_TOS_DATE
    setting, 374
LEGAL_URL
    setting, 375
LICENSE_EXTRA
    setting, 375
LICENSE_FILTER
    setting, 375
LICENSE_REQUIRED
    setting, 376
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
    setting, 376
list_languages
    weblate admin command, 413
list_translators
    weblate admin command, 413
list_versions
    weblate admin command, 414
list-components
    opción de línea de comando wlc, 167
list-languages
    opción de línea de comando wlc, 167
list-projects
    opción de línea de comando wlc, 167
list-translations
    opción de línea de comando wlc, 167
load() (*método de wlc.config.WeblateConfig*), 171
loadpo
    weblate admin command, 414
LOCALIZE_CDN_PATH
    setting, 376
LOCALIZE_CDN_URL
    setting, 376
lock
    opción de línea de comando wlc, 168
lock_translation
    weblate admin command, 414
lock-status
    opción de línea de comando wlc, 168
LOGIN_REQUIRED_URLS
    setting, 376
LOGIN_REQUIRED_URLS_EXCEPTIONS
    setting, 377
ls
    opción de línea de comando wlc, 167

M

main() (*en el módulo wlc.main*), 172
MATOMO_SITE_ID
    setting, 377
MATOMO_URL
    setting, 377
mi18n lang
    file format, 96
migrate
    weblate admin command, 414
monolingual
    translation, 108
move_language
    weblate admin command, 415
módulo
    wlc, 171
    wlc.config, 171
    wlc.main, 172

N

NEARBY_MESSAGES
    setting, 378

O

opción de línea de comando add_suggestions
    --author, 406
opción de línea de comando auto_translate
    --add, 406
    --inconsistent, 406
    --mode, 406
    --mt, 406
    --overwrite, 406
    --source, 406
    --threshold, 406
    --user, 406
opción de línea de comando commit_pending
    --age, 407
opción de línea de comando createadmin
    --email, 408
    --name, 408
    --no-password, 408
    --password, 408
    --update, 408
    --username, 408
opción de línea de comando import_json
    --ignore, 409
    --main-component, 409
    --project, 409
    --update, 409
opción de línea de comando import_memory
    --language-map, 410
opción de línea de comando import_project
    --base-file-template, 410
    --file-format, 410
    --language-regex, 410
    --license, 411
    --license-url, 411
    --main-component, 410
    --name-template, 410
    --new-base-template, 410
    --vcs, 411
opción de línea de comando importusers
    --check, 412

opción de línea de comando install_addon
--addon, 412
--configuration, 412
--update, 412
opción de línea de comando install_machinery
--configuration, 413
--service, 413
--update, 413
opción de línea de comando list_translators
--language-code, 413
opción de línea de comando loadpo
--force, 414
--lang, 414
opción de línea de comando pushgit
--force-commit, 415
opción de línea de comando setupgroups
--no-privs-update, 416
--no-projects-update, 416
opción de línea de comando setuplang
--no-update, 416
opción de línea de comando wlc
--author-email, 169
--author-name, 169
--config, 167
--config-section, 167
--convert, 168
--format, 167
--fuzzy, 168
--input, 168
--key, 167
--method, 168
--output, 168
--overwrite, 168
--url, 167
changes, 168
cleanup, 168
commit, 167
download, 168
list-components, 167
list-languages, 167
list-projects, 167
list-translations, 167
lock, 168
lock-status, 168
ls, 167
pull, 167
push, 167
repo, 168
reset, 167
show, 167
stats, 168
unlock, 168
upload, 168
version, 167

P

PAGURE_CREDENTIALS
setting, 378
PHP strings
file format, 97
PIWIK_SITE_ID
setting, 377
PIWIK_URL
setting, 377
PO
file format, 88
post() (*método de wlc.Weblate*), 171
post_add() (*método de weblate.addons.base.BaseAddon*), 451
post_commit() (*método de weblate.addons.base.BaseAddon*), 451
post_push() (*método de weblate.addons.base.BaseAddon*), 451
post_update() (*método de weblate.addons.base.BaseAddon*), 451
POSTGRES_DATABASE, 178
pre_commit() (*método de weblate.addons.base.BaseAddon*), 452
pre_push() (*método de weblate.addons.base.BaseAddon*), 452
pre_update() (*método de weblate.addons.base.BaseAddon*), 452
PRIVACY_URL
setting, 379
PRIVATE_COMMIT_EMAIL_OPT_IN
setting, 379
PRIVATE_COMMIT_EMAIL_TEMPLATE
setting, 379
PROJECT_BACKUP_KEEP_COUNT
setting, 379
PROJECT_BACKUP_KEEP_DAYS
setting, 380
PROJECT_NAME_RESTRICT_RE
setting, 380
PROJECT_WEB_RESTRICT_HOST
setting, 380
PROJECT_WEB_RESTRICT_NUMERIC
setting, 380
PROJECT_WEB_RESTRICT_RE
setting, 380
pull
opción de línea de comando wlc, 167
push
opción de línea de comando wlc, 167
pushgit
weblate admin command, 415
Python, 170
Python Enhancement Proposals
PEP 484, 447, 462

Q

Qt
file format, 98

R

RATELIMIT_ATTEMPTS
    setting, 381
RATELIMIT_LOCKOUT
    setting, 381
RATELIMIT_WINDOW
    setting, 381
RC
    file format, 102
REDIS_PASSWORD, 193
register_command() (*en el módulo wlc.main*), 172
REGISTRATION_ALLOW_BACKENDS
    setting, 381
REGISTRATION_CAPTCHA
    setting, 382
REGISTRATION_EMAIL_MATCH
    setting, 382
REGISTRATION_OPEN
    setting, 382
REGISTRATION_REBIND
    setting, 382
repo
    opción de línea de comando wlc, 168
REPOSITORY_ALERT_THRESHOLD
    setting, 382
REQUIRE_LOGIN
    setting, 383
reset
    opción de línea de comando wlc, 167
ResourceDictionary
    file format, 98
REST, 117
RESX
    file format, 99
RFC
    RFC 5646, 82
Ruby YAML
    file format, 99
Ruby YAML Ain't Markup Language
    file format, 99

S

save_state() (*método de weblate.addons.base.BaseAddon*), 452
SECRET_KEY
    setting, 224
SENTRY_DSN
    setting, 383
SERVER_EMAIL
    setting, 224
SESSION_COOKIE_AGE_AUTHENTICATED
    setting, 383
SESSION_ENGINE
    setting, 224
setting
    ADMINS, 223
    AKISMET_API_KEY, 360
    ALLOWED_HOSTS, 223
    ANONYMOUS_USER_NAME, 360
    AUDITLOG_EXPIRY, 360
    AUTH_LOCK_ATTEMPTS, 360
    AUTH_TOKEN_VALID, 361
    AUTO_UPDATE, 361
    AUTOFIX_LIST, 362
    AVATAR_URL_PREFIX, 361
    BACKGROUND_TASKS, 362
    BASIC_LANGUAGES, 363
    BITBUCKETSERVER_CREDENTIALS, 372
    BORG_EXTRA_ARGS, 363
    CACHE_DIR, 363
    CHECK_LIST, 364
    COMMENT_CLEANUP_DAYS, 364
    COMMIT_PENDING_HOURS, 365
    CONTACT_FORM, 365
    CSP_CONNECT_SRC, 364
    CSP_FONT_SRC, 364
    CSP_IMG_SRC, 364
    CSP_SCRIPT_SRC, 364
    CSP_STYLE_SRC, 364
    DATA_DIR, 365
    DATABASE_BACKUP, 366
    DATABASES, 224
    DEBUG, 224
    DEFAULT_ACCESS_CONTROL, 366
    DEFAULT_ADD_MESSAGE, 367
    DEFAULT_ADDON_MESSAGE, 367
    DEFAULT_ADDONS, 367
    DEFAULT_AUTO_WATCH, 366
    DEFAULT_COMMIT_MESSAGE, 367
    DEFAULT_COMMITER_EMAIL, 367
    DEFAULT_COMMITER_NAME, 368
    DEFAULT_DELETE_MESSAGE, 367
    DEFAULT_FROM_EMAIL, 224
    DEFAULT_LANGUAGE, 368
    DEFAULT_MERGE_MESSAGE, 367
    DEFAULT_MERGE_STYLE, 368
    DEFAULT_PAGE_LIMIT, 378
    DEFAULT_PULL_MESSAGE, 368
    DEFAULT_RESTRICTED_COMPONENT, 367
    DEFAULT_SHARED_TM, 368
    DEFAULT_TRANSLATION_PROPAGATION, 368
    ENABLE_AVATARS, 369
    ENABLE_HOOKS, 369
    ENABLE_HTTPS, 369
    ENABLE_SHARING, 369
    EXTRA_HTML_HEAD, 369
    GET_HELP_URL, 369
    GITEA_CREDENTIALS, 370
    GITHUB_CREDENTIALS, 371
    GITLAB_CREDENTIALS, 370
    GOOGLE_ANALYTICS_ID, 373
    HIDE_REPO_CREDENTIALS, 373
    HIDE_VERSION, 373
    INTERLEDGER_PAYMENT_POINTERS, 373
    IP_BEHIND_REVERSE_PROXY, 374

IP_PROXY_HEADER, 374
IP_PROXY_OFFSET, 374
LEGAL_TOS_DATE, 374
LEGAL_URL, 375
LICENSE_EXTRA, 375
LICENSE_FILTER, 375
LICENSE_REQUIRED, 376
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 376
LOCALIZE_CDN_PATH, 376
LOCALIZE_CDN_URL, 376
LOGIN_REQUIRED_URLS, 376
LOGIN_REQUIRED_URLS_EXCEPTIONS, 377
MATOMO_SITE_ID, 377
MATOMO_URL, 377
NEARBY_MESSAGES, 378
PAGURE_CREDENTIALS, 378
PIWIK_SITE_ID, 377
PIWIK_URL, 377
PRIVACY_URL, 379
PRIVATE_COMMIT_EMAIL_OPT_IN, 379
PRIVATE_COMMIT_EMAIL_TEMPLATE, 379
PROJECT_BACKUP_KEEP_COUNT, 379
PROJECT_BACKUP_KEEP_DAYS, 380
PROJECT_NAME_RESTRICT_RE, 380
PROJECT_WEB_RESTRICT_HOST, 380
PROJECT_WEB_RESTRICT_NUMERIC, 380
PROJECT_WEB_RESTRICT_RE, 380
RATELIMIT_ATTEMPTS, 381
RATELIMIT_LOCKOUT, 381
RATELIMIT_WINDOW, 381
REGISTRATION_ALLOW_BACKENDS, 381
REGISTRATION_CAPTCHA, 382
REGISTRATION_EMAIL_MATCH, 382
REGISTRATION_OPEN, 382
REGISTRATION_REBIND, 382
REPOSITORY_ALERT_THRESHOLD, 382
REQUIRE_LOGIN, 383
SECRET_KEY, 224
SENTRY_DSN, 383
SERVER_EMAIL, 224
SESSION_COOKIE_AGE_AUTHENTICATED, 383
SESSION_ENGINE, 224
SIMPLIFY_LANGUAGES, 383
SINGLE_PROJECT, 384
SITE_DOMAIN, 383
SITE_TITLE, 384
SPECIAL_CHARS, 384
SSH_EXTRA_ARGS, 384
STATUS_URL, 385
SUGGESTION_CLEANUP_DAYS, 385
UNUSED_ALERT_DAYS, 385
UPDATE_LANGUAGES, 385
URL_PREFIX, 385
VCS_API_DELAY, 386
VCS_BACKENDS, 386
VCS_CLONE_DEPTH, 386
WEBLATE_ADDONS, 387
WEBLATE_EXPORTERS, 388
WEBLATE_FORMATS, 388
WEBLATE_GPG_IDENTITY, 388
WEBLATE_MACHINERY, 388
WEBSITE_REQUIRED, 388
setupgroups
weblate admin command, 416
setuplang
weblate admin command, 416
show
opción de línea de comando wlc, 167
SIMPLIFY_LANGUAGES
setting, 383
SINGLE_PROJECT
setting, 384
SITE_DOMAIN
setting, 383
SITE_TITLE
setting, 384
SPECIAL_CHARS
setting, 384
SSH_EXTRA_ARGS
setting, 384
stats
opción de línea de comando wlc, 168
STATUS_URL
setting, 385
store_post_load() (*método de weblate.addons.base.BaseAddon*), 452
string resources
file format, 82
SUGGESTION_CLEANUP_DAYS
setting, 385

T

translation
bilingual, 108
iOS, 83
monolingual, 108
TS
file format, 98

U

unit_pre_create() (*método de weblate.addons.base.BaseAddon*), 452
unlock
opción de línea de comando wlc, 168
unlock_translation
weblate admin command, 415
UNUSED_ALERT_DAYS
setting, 385
UPDATE_LANGUAGES
setting, 385
updatechecks
weblate admin command, 416
updategit
weblate admin command, 416

upload
    opción de línea de comando wlc, 168
URL_PREFIX
    setting, 385

V

variables de entorno
    CELERY_BACKUP_OPTIONS, 178, 196, 197
    CELERY_BEAT_OPTIONS, 178, 196, 197
    CELERY_MAIN_OPTIONS, 178, 196
    CELERY_MEMORY_OPTIONS, 178, 196, 197
    CELERY_NOTIFY_OPTIONS, 178, 196, 197
    CELERY_TRANSLATE_OPTIONS, 178, 196, 197
    CLIENT_MAX_BODY_SIZE, 186
    POSTGRES_ALTER_ROLE, 192
    POSTGRES_CONN_MAX_AGE, 192
    POSTGRES_DATABASE, 178, 191
    POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 192
    POSTGRES_HOST, 191
    POSTGRES_PASSWORD, 191
    POSTGRES_PASSWORD_FILE, 191
    POSTGRES_PORT, 191
    POSTGRES_SSL_MODE, 191
    POSTGRES_USER, 191
    REDIS_DB, 192
    REDIS_HOST, 192
    REDIS_PASSWORD, 193
    REDIS_PASSWORD_FILE, 193
    REDIS_PORT, 192
    REDIS_TLS, 193
    REDIS_VERIFY_SSL, 193
    ROLLBAR_ENVIRONMENT, 195
    ROLLBAR_KEY, 195
    SENTRY_DSN, 195
    SENTRY_ENVIRONMENT, 195
    SENTRY_PROFILES_SAMPLE_RATE, 195
    SENTRY_TRACES_SAMPLE_RATE, 195
    SOCIAL_AUTH_SLACK_SECRET, 190
    WEB_WORKERS, 178, 196, 197
    WEBLATE_ADD_ADDONS, 196
    WEBLATE_ADD_APPS, 196
    WEBLATE_ADD_AUTOFIX, 196
    WEBLATE_ADD_CHECK, 196
    WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 182
    WEBLATE_ADMIN_EMAIL, 178--180
    WEBLATE_ADMIN_NAME, 178--180
    WEBLATE_ADMIN_PASSWORD, 174, 178--180
    WEBLATE_ADMIN_PASSWORD_FILE, 180
    WEBLATE_AKISMET_API_KEY, 184, 423
    WEBLATE_ALLOWED_HOSTS, 180, 223, 228, 384
    WEBLATE_API_RATELIMIT_ANON, 120, 185
    WEBLATE_API_RATELIMIT_USER, 120, 185
    WEBLATE_AUTH_LDAP_BIND_DN, 187
    WEBLATE_AUTH_LDAP_BIND_PASSWORD, 187
    WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE, 187
    WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS, 187
    WEBLATE_AUTH_LDAP_SERVER_URI, 187
    WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 187
    WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 187
    WEBLATE_AUTH_LDAP_USER_SEARCH, 187
    WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 187
    WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 187
    WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 187
    WEBLATE_AUTO_UPDATE, 194
    WEBLATE_AVATAR_URL_PREFIX, 185
    WEBLATE_BASIC_LANGUAGES, 184
    WEBLATE_BITBUCKETSERVER_HOST, 183
    WEBLATE_BITBUCKETSERVER_TOKEN, 183
    WEBLATE_BITBUCKETSERVER_USERNAME, 183
    WEBLATE_BORG_EXTRA_ARGS, 185
    WEBLATE_CONTACT_FORM, 180
    WEBLATE_CORS_ALLOWED_ORIGINS, 186
    WEBLATE_CSP_CONNECT_SRC, 184
    WEBLATE_CSP_FONT_SRC, 184
    WEBLATE_CSP_IMG_SRC, 184
    WEBLATE_CSP_SCRIPT_SRC, 184
    WEBLATE_CSP_STYLE_SRC, 184
    WEBLATE_DATABASE_BACKUP, 192
    WEBLATE_DEBUG, 179, 247
    WEBLATE_DEFAULT_ACCESS_CONTROL, 183
    WEBLATE_DEFAULT_AUTO_WATCH, 185
    WEBLATE_DEFAULT_COMMITER_EMAIL, 184
    WEBLATE_DEFAULT_COMMITER_NAME, 184
    WEBLATE_DEFAULT_FROM_EMAIL, 180
    WEBLATE_DEFAULT_PULL_MESSAGE, 183
    WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 184
    WEBLATE_DEFAULT_SHARED_TM, 184
    WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 184
    WEBLATE_EMAIL_BACKEND, 194
    WEBLATE_EMAIL_HOST, 193
    WEBLATE_EMAIL_HOST_PASSWORD, 193, 194
    WEBLATE_EMAIL_HOST_PASSWORD_FILE, 194
    WEBLATE_EMAIL_HOST_USER, 193
    WEBLATE_EMAIL_PORT, 193, 194
    WEBLATE_EMAIL_USE_SSL, 193, 194
    WEBLATE_EMAIL_USE_TLS, 193, 194
    WEBLATE_ENABLE_AVATARS, 185
    WEBLATE_ENABLE_HOOKS, 185
    WEBLATE_ENABLE_HTTPS, 181, 257

WEBLATE_ENABLE_SHARING, 185
WEBLATE_EXTRA_HTML_HEAD, 186
WEBLATE_GET_HELP_URL, 195
WEBLATE_GITEA_HOST, 183
WEBLATE_GITEA_TOKEN, 183
WEBLATE_GITEA_USERNAME, 183
WEBLATE_GITHUB_HOST, 182, 246
WEBLATE_GITHUB_TOKEN, 182
WEBLATE_GITHUB_USERNAME, 182
WEBLATE_GITLAB_HOST, 183
WEBLATE_GITLAB_TOKEN, 183
WEBLATE_GITLAB_USERNAME, 183
WEBLATE_GOOGLE_ANALYTICS_ID, 182
WEBLATE_GPG_IDENTITY, 184
WEBLATE_HIDE_VERSION, 184
WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 181
WEBLATE_IP_PROXY_HEADER, 182
WEBLATE_LEGAL_URL, 195
WEBLATE_LICENSE_FILTER, 184
WEBLATE_LICENSE_REQUIRED, 184
WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 185
WEBLATE_LOCALIZE_CDN_PATH, 195
WEBLATE_LOCALIZE_CDN_URL, 195
WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 182
WEBLATE_LOGLEVEL, 179, 448
WEBLATE_LOGLEVEL_DATABASE, 179
WEBLATE_NO_EMAIL_AUTH, 191
WEBLATE_PAGURE_HOST, 183
WEBLATE_PAGURE_TOKEN, 183
WEBLATE_PAGURE_USERNAME, 183
WEBLATE_PRIVACY_URL, 195
WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN, 186
WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE, 186
WEBLATE_RATELIMIT_ATTEMPTS, 185, 425
WEBLATE_RATELIMIT_LOCKOUT, 185
WEBLATE_RATELIMIT_WINDOW, 185
WEBLATE_REGISTRATION_ALLOW_BACKENDS, 181
WEBLATE_REGISTRATION_OPEN, 180
WEBLATE_REGISTRATION_REBIND, 181
WEBLATE_REMOVE_ADDONS, 196
WEBLATE_REMOVE_APPS, 196
WEBLATE_REMOVE_AUTOFIX, 196
WEBLATE_REMOVE_CHECK, 196
WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 182
WEBLATE_REQUIRE_LOGIN, 182, 383
WEBLATE_SAML_ID_ATTR_EMAIL, 191
WEBLATE_SAML_ID_ATTR_NAME, 191
WEBLATE_SAML_ID_ATTR_USER_PERMANENT_ID, 191
WEBLATE_SAML_ID_ATTR_USERNAME, 191
WEBLATE_SAML_IDP_ENTITY_ID, 191
WEBLATE_SAML_IDP_IMAGE, 191
WEBLATE_SAML_IDP_TITLE, 191
WEBLATE_SAML_IDP_URL, 191
WEBLATE_SAML_IDP_X509CERT, 191
WEBLATE_SECURE_PROXY_SSL_HEADER, 181, 182
WEBLATE_SERVER_EMAIL, 180
WEBLATE_SERVICE, 179, 197
WEBLATE_SILENCED_SYSTEM_CHECKS, 184, 254
WEBLATE_SIMPLIFY_LANGUAGES, 183
WEBLATE_SITE_DOMAIN, 179, 195, 225, 243, 384
WEBLATE_SITE_TITLE, 179
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 189
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 189
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 189
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECR
189
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENA
189
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 188
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY, 188
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET, 188
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 188
WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 189
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 189
WEBLATE_SOCIAL_AUTH_FEDORA, 190
WEBLATE_SOCIAL_AUTH_GITEA_API_URL, 189
WEBLATE_SOCIAL_AUTH_GITEA_KEY, 189
WEBLATE_SOCIAL_AUTH_GITEA_SECRET, 189
WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_API_URL, 188
WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_KEY, 188
WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_SCOPE, 188
WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_SECRET, 188
WEBLATE_SOCIAL_AUTH_GITHUB_ENTERPRISE_URL, 188
WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 188
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY, 188
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME, 188
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET,

188
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 188
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID, 188
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY, 188
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET, 188
WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 189
WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 189
WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 189
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 189
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 189
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 189
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 189
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 190
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 190
WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 190
WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE, 190
WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 190
WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 190
WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 190
WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE, 190
WEBLATE_SOCIAL_AUTH_OIDC_KEY, 190
WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT, 190
WEBLATE_SOCIAL_AUTH_OIDC_SECRET, 190
WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY, 190
WEBLATE_SOCIAL_AUTH_OPENINFRA, 190
WEBLATE_SOCIAL_AUTH_OPENSUSE, 190
WEBLATE_SOCIAL_AUTH_SLACK_KEY, 190
WEBLATE_SOCIAL_AUTH_UBUNTU, 190
WEBLATE_SSH_EXTRA_ARGS, 185
WEBLATE_STATUS_URL, 195
WEBLATE_TIME_ZONE, 181
WEBLATE_UNUSED_ALERT_DAYS, 186
WEBLATE_URL_PREFIX, 184
WEBLATE_WEBSITE_REQUIRED, 184
WEBLATE_WORKERS, 178, 196, 197
WL_BRANCH, 357
WL_COMPONENT_NAME, 357
WL_COMPONENT_SLUG, 357
WL_COMPONENT_URL, 357
WL_ENGAGE_URL, 357
WL_FILE_FORMAT, 357
WL_FILEMASK, 357
WL_LANGUAGE, 357
WL_NEW_BASE, 357
WL_PATH, 356
WL_PREVIOUS_HEAD, 357
WL_PROJECT_NAME, 357
WL_PROJECT_SLUG, 357
WL_REPO, 356
WL_TEMPLATE, 357
WL_VCS, 356
WORKS_API_DELAY
setting, 386
WORKS_BACKENDS
setting, 386
WORKS_CLONED_DOMAINS
setting, 386
WORKS_LISTED_EMAILS
opción de línea de comando wlc, 167

W

WEB_WORKERS, 178, 196
Weblate (*clase en wlc*), 171
weblate-admin command
add_suggestions, 406
auto_translate, 406
celery_queues, 407
checkgit, 407
cleanup_ssh_keys, 408
cleanuptrans, 407
commit_pending, 407
commitgit, 407
createadmin, 408
dump_memory, 408
dumpuserdata, 408
import_demo, 409
import_json, 409
import_memory, 410
import_project, 410
importuserdata, 412
importusers, 412
install_addon, 412
install_machinery, 413
list_languages, 413
list_translators, 413
list_versions, 414
loadpo, 414
lock_translation, 414
migrate, 414
move_language, 415
pushgit, 415
setupgroups, 416
setuplang, 416
unlock_translation, 415
updatechecks, 416

updategit, 416
WEBLATE_ADDONS
setting, 387
WEBLATE_ADMIN_EMAIL, 178, 180
WEBLATE_ADMIN_NAME, 178, 180
WEBLATE_ADMIN_PASSWORD, 174, 178--180
WEBLATE_ADMIN_PASSWORD_FILE, 180
WEBLATE_AKISMET_API_KEY, 423
WEBLATE_ALLOWED_HOSTS, 223, 228, 384
WEBLATE_API_RATELIMIT_ANON, 120
WEBLATE_API_RATELIMIT_USER, 120
WEBLATE_AUTH_LDAP_BIND_PASSWORD, 187
WEBLATE_DEBUG, 179, 247
WEBLATE_EMAIL_HOST_PASSWORD, 194
WEBLATE_EMAIL_PORT, 193, 194
WEBLATE_EMAIL_USE_SSL, 193, 194
WEBLATE_EMAIL_USE_TLS, 193, 194
WEBLATE_ENABLE_HTTPS, 257
WEBLATE_EXPORTERS
setting, 388
WEBLATE_FORMATS
setting, 388
WEBLATE_GITHUB_HOST, 246
WEBLATE_GPG_IDENTITY
setting, 388
WEBLATE_LOCALIZE_CDN_PATH, 195
WEBLATE_LOGLEVEL, 448
WEBLATE_MACHINERY
setting, 388
WEBLATE_RATELIMIT_ATTEMPTS, 425
WEBLATE_REQUIRE_LOGIN, 383
WEBLATE_SECURE_PROXY_SSL_HEADER, 181
WEBLATE_SERVICE, 179
WEBLATE_SILENCED_SYSTEM_CHECKS, 254
WEBLATE_SITE_DOMAIN, 195, 225, 243, 384
WEBLATE_WORKERS, 178, 197
WeblateConfig (*clase en wlc.config*), 171
WeblateException, 171
WEBSITE_REQUIRED
setting, 388
wlc, 165
módulo, 171
wlc.config
módulo, 171
wlc.main
módulo, 172
WPF
file format, 98

X

XLIFF
file format, 103
XML
file format, 87

Y

YAML
file format, 105
YAML Ain't Markup Language
file format, 105