



The Weblate Manual

Versión 4.6

Michal Čihař

19 de abril de 2021

1	Documentación para usuarios	1
1.1	Fundamentos de Weblate	1
1.2	Registro y perfil de usuario	1
1.3	Traducir con Weblate	10
1.4	Descargar y cargar traducciones	19
1.5	Glosario	22
1.6	Comprobaciones y correcciones	25
1.7	Búsquedas	42
1.8	Flujos de trabajo de traducción	47
1.9	Preguntas frecuentes	50
1.10	Formatos de archivo admitidos	58
1.11	Integración de control de versiones	77
1.12	API REST de Weblate	84
1.13	Cliente de Weblate	127
1.14	API de Python de Weblate	132
2	Documentación para administradores	135
2.1	Instrucciones de configuración	135
2.2	Implantaciones de Weblate	194
2.3	Actualizar Weblate	195
2.4	Respalidar y trasladar Weblate	201
2.5	Autenticación	207
2.6	Control de acceso	216
2.7	Proyectos de traducción	225
2.8	Language definitions	242
2.9	Regionalización continua	245
2.10	Licensing translations	255
2.11	Proceso de traducción	256
2.12	Comprobaciones y correcciones	262
2.13	Traducción automática	270
2.14	Complementos	276
2.15	Memoria de traducción	287
2.16	Configuración	289
2.17	Configuración de muestra	317
2.18	Órdenes de gestión	332
2.19	Anuncios	343
2.20	Listas de componentes	345
2.21	Módulos opcionales de Weblate	346
2.22	Personalizar Weblate	351
2.23	Interfaz de gestión	353
2.24	Obtener ayuda con Weblate	361

2.25	Documentos legales	364
3	Documentación para contribuidores	366
3.1	Contribuir con Weblate	366
3.2	Empezar a contribuir con código a Weblate	368
3.3	Código fuente de Weblate	372
3.4	Depurar Weblate	373
3.5	Weblate por dentro	374
3.6	Desarrollo de complementos	376
3.7	Interfaz de Weblate	378
3.8	Informar de problemas en Weblate	379
3.9	Weblate testsuite and continuous integration	379
3.10	Esquemas de datos	381
3.11	Releasing Weblate	385
3.12	Seguridad y privacidad	386
3.13	Acerca de Weblate	386
3.14	Licencia	387
4	Historial de cambios	388
4.1	Weblate 4.6	388
4.2	Weblate 4.5.3	389
4.3	Weblate 4.5.2	389
4.4	Weblate 4.5.1	389
4.5	Weblate 4.5	390
4.6	Weblate 4.4.2	391
4.7	Weblate 4.4.1	391
4.8	Weblate 4.4	391
4.9	Weblate 4.3.2	392
4.10	Weblate 4.3.1	392
4.11	Weblate 4.3	393
4.12	Weblate 4.2.2	394
4.13	Weblate 4.2.1	394
4.14	Weblate 4.2	394
4.15	Weblate 4.1.1	395
4.16	Weblate 4.1	395
4.17	Weblate 4.0.4	396
4.18	Weblate 4.0.3	397
4.19	Weblate 4.0.2	397
4.20	Weblate 4.0.1	397
4.21	Weblate 4.0	398
4.22	Serie 3.x de Weblate	398
4.23	Serie 2.x de Weblate	410
4.24	Serie 1.x de Weblate	421
4.25	Serie 0.x de Weblate	425
	Índice de Módulos Python	429
	HTTP Routing Table	430
	Índice	433

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.

Nuevo en la versión 2.5.

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 screenshot shows the Weblate web application interface. At the top is a dark navigation bar with the Weblate logo, a 'Dashboard' link, and dropdown menus for 'Projects', 'Languages', and 'Checks'. On the right of this bar are icons for settings, a plus sign, a user profile, and a menu. Below the navigation bar, on the left, is a user profile icon and the text 'Your profile'. A horizontal menu contains links for 'Languages', 'Preferences' (which is highlighted with a dark green background), 'Notifications', 'Account', 'Profile', 'Licenses', 'Audit log', and 'API access'. The 'Preferences' panel is open, showing a title bar with 'Preferences' and a help icon. The panel contains several settings: a checkbox for 'Hide completed translations on the dashboard' (unchecked); a 'Translation editor mode' dropdown set to 'Full editor'; a 'Zen editor mode' dropdown set to 'Top to bottom'; a 'Number of nearby strings' input field set to '15' with a descriptive text below it; a checked checkbox for 'Show secondary translations in the Zen mode'; an unchecked checkbox for 'Hide source if a secondary translation exists'; an 'Editor link' section with an empty input field and descriptive text; a 'Special characters' section with an empty input field and descriptive text; and a 'Default dashboard view' section with two radio buttons, 'Watched translations' (selected) and 'Suggested translations'. At the bottom of the panel is a dark 'Save' button.

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

The menu has these options:

- *Projects > Browse all projects* in the main menu showing translation status for each project on the Weblate instance.

- Selecting a language in the main menu *Languages* will show translation status of all projects, filtered by one of your primary languages.
- *Watched translations* in the Dashboard will show translation status of only those projects you are watching, filtered by your primary languages.

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

[Dashboard](#)

[Projects](#) ▾

[Languages](#) ▾

[Checks](#) ▾









 Dashboard

Watched translations 13

Suggested translations 5

Insights ▾

Search



Component	Translated	Untranslated	Untranslated words	Checks	Suggestions
 WeblateOrg/Android — Czech   MIT	76%	3	3		
 WeblateOrg/Django — Hungarian   GPL-3.0	69%	8	109	1	
 WeblateOrg/Django — Czech   GPL-3.0	96%	1	12	4	
 WeblateOrg/Django — Hebrew   GPL-3.0	92%	2	15		
 WeblateOrg/Djangojs — Czech   GPL-3.0	✓				
 WeblateOrg/Djangojs — Hebrew   GPL-3.0	✓				
 WeblateOrg/Djangojs — Hungarian   GPL-3.0	96%	2	6		
 WeblateOrg/Language names — Czech   GPL-3.0	✓				
 WeblateOrg/Language names — Hungarian   GPL-3.0	81%	4	5		
 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 4.6

[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:

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb trail shows 'WeblateOrg / Django / Czech / Translate'. A status bar indicates 'translated 96%'. The main area shows a string list with '1 / 26' items. The selected string is 'Soubory' in English, which has been translated to 'קבצים' in Czech. The interface includes buttons for 'Save', 'Suggest', and 'Skip'. A sidebar on the right contains a 'Glossary' section, a 'String information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

Translation Editor:

- Hebrew:
- English: Files
- Czech: Soubory

Buttons: Save, Suggest, Skip

String List:

Language	Translated string
Hebrew	קבצים
Hungarian	Fájlok
English	Files

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:** 23 seconds ago
- Source string age:** 24 seconds ago
- Translation file:** `weblate/locale/cs/LC_MESSAGES/django.po`, string 1

1.2.5 Preferencias

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](#).

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.

Powered by [Weblate 4.6](#) [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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

 Your profile

[Languages](#) [Preferences](#) [Notifications](#) **Account** [Profile](#) [Licenses](#) [Audit log](#) [API access](#)

Account

Username

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

Full name

E-mail

You can add another e-mail address below.

Your name and e-mail will appear as commit authorship.

Save

Current user identities

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

Add new association

 **E-mail**

Removal

Account removal deletes all your private data.

Remove my account

User data

You can download all your private data.

Download user data

1.2.8 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.

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 públicamente y cualquiera puede contribuir
- Visible solo a un grupo determinado de traductores

Ver también:

Control de acceso, Flujos de trabajo de traducción

1.3.1 Proyectos de traducción

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

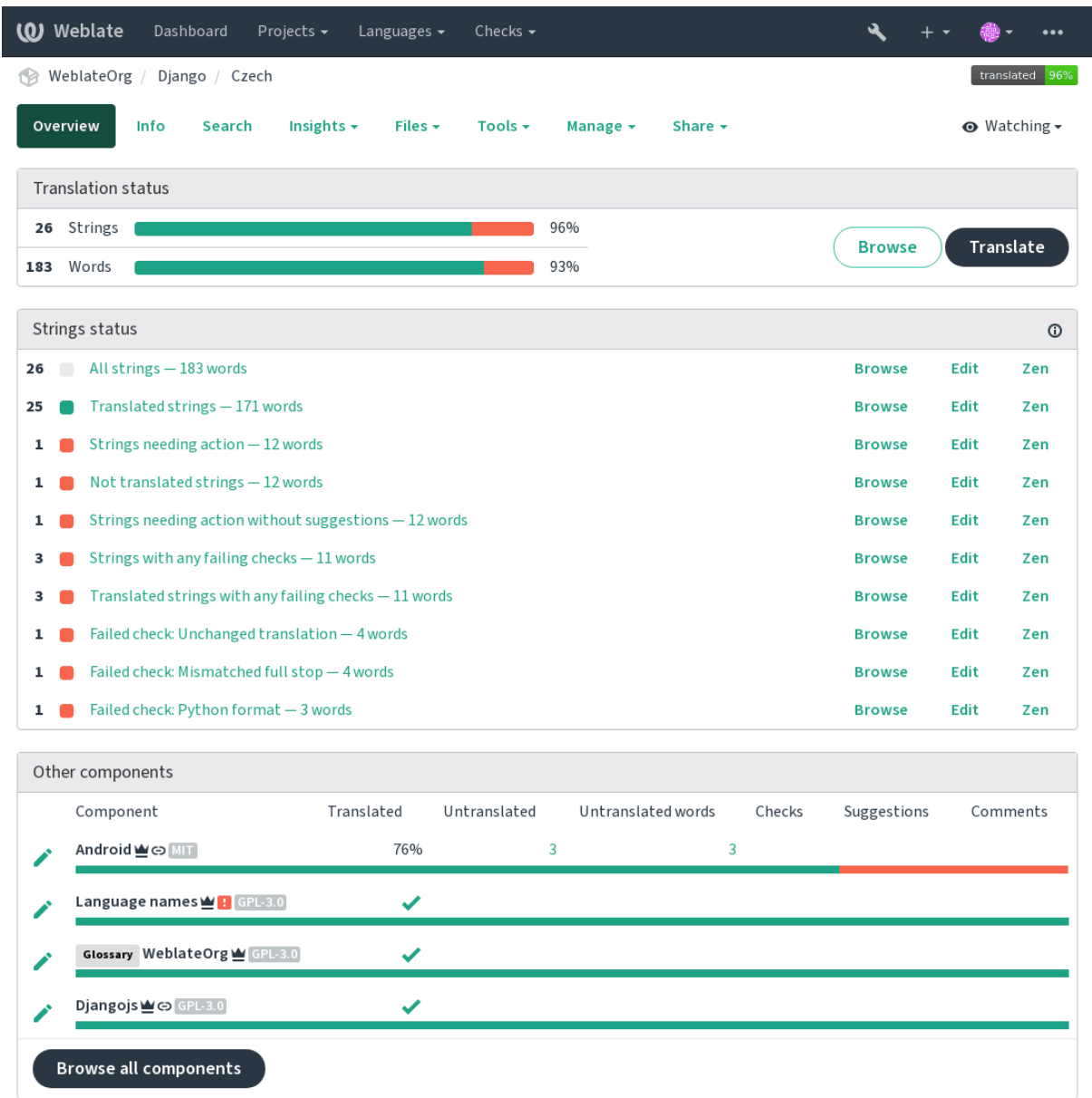
Component	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Android	79%	30	30	3		
Language names	95%	4	5			
Glossary WeblateOrg	100%	0	0	0	0	

Add new translation component

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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 *Not translated strings* or *Strings needing action*. 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.



Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.3.3 Sugerencias

Nota: Actual permissions might vary depending on your Weblate configuration.

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

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).

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.

Bajo la traducción, los traductores verán las sugerencias hechas por otros pendientes de ser aceptadas (✓), aceptadas con cambios (🔗) o eliminadas (🗑️).

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
Dashboard
Projects
Languages
Checks

WeblateOrg / Django / Czech / Translate
translated 96%

1 / 1
Custom search
'%(count)s word'
Zen

Translation

English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
%(count)s slovo
Czech, Few
%(count)s slova
Czech, Other
%(count)s slov
Plural formula: (n==1)?0:(n>=2 && n<=4)?1:2
Needs editing
Save
Suggest
Skip

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
16 seconds ago
Source string age
16 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 4.6
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Atajos de teclado

Distinto en la versión 2.18: Se han renovado los atajos de teclado en la versión 2.18 para hacer menos probable que entren en conflicto con los atajos predeterminados del navegador o del sistema.

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 págo Ctrl+↑ o Alt+↑ o ⌘+↑	Navegar a la traducción anterior de la búsqueda actual.
Alt+Av págo Ctrl+↓ o Alt+↓ o ⌘+↓	Navegar a la traducción siguiente de la búsqueda actual.
Alt+Intro o Ctrl+Intro o ⌘+Intro	Guardar la traducción actual.
Ctrl+Mayús+Intro o ⌘+Mayús+Intro	Unmark translation as needing edit and submit it.
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 <i>Sugerencias automáticas</i> ; vea <i>Sugerencias automáticas</i> .
Ctrl+1 to Ctrl+9 or Cmd+1 to Cmd+9	Copies placeable of given number from source string.
Ctrl+M+1 to 9 or Cmd+M+1 to 9	Copy the machine translation of given number to current translation.
Ctrl+I+1 to 9 or Cmd+I+1 to 9	Ignorar un elemento en la lista de comprobaciones fallidas.
Ctrl+J or Cmd+J	Muestra la pestaña <i>Cadenas cercanas</i> .
Ctrl+S or Cmd+S	Focus search field.
Ctrl+O or Cmd+O	Copiar la cadena de origen.
Ctrl+Y or Cmd+Y	Toggle the <i>Needs editing</i> flag.

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:

- Caracteres configurados por cada usuario en el *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*

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Hebrew' language. The main area shows a translation for the string 'Files' (English) to 'קבצים' (Hebrew). The interface includes a top navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. A breadcrumb trail shows 'WeblateOrg / Django / Hebrew / Translate'. A progress indicator shows 'translated 92%'. The main area has a 'Translation' section with a 'Needs editing' checkbox and buttons for 'Save', 'Suggest', and 'Skip'. Below this is a table of 'Nearby strings' in other languages: Czech ('Soubory'), Hungarian ('Fájlok'), and English ('Files'). On the right, there are panels for 'Glossary', 'String information' (including 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'), and a 'Needs editing' checkbox.

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

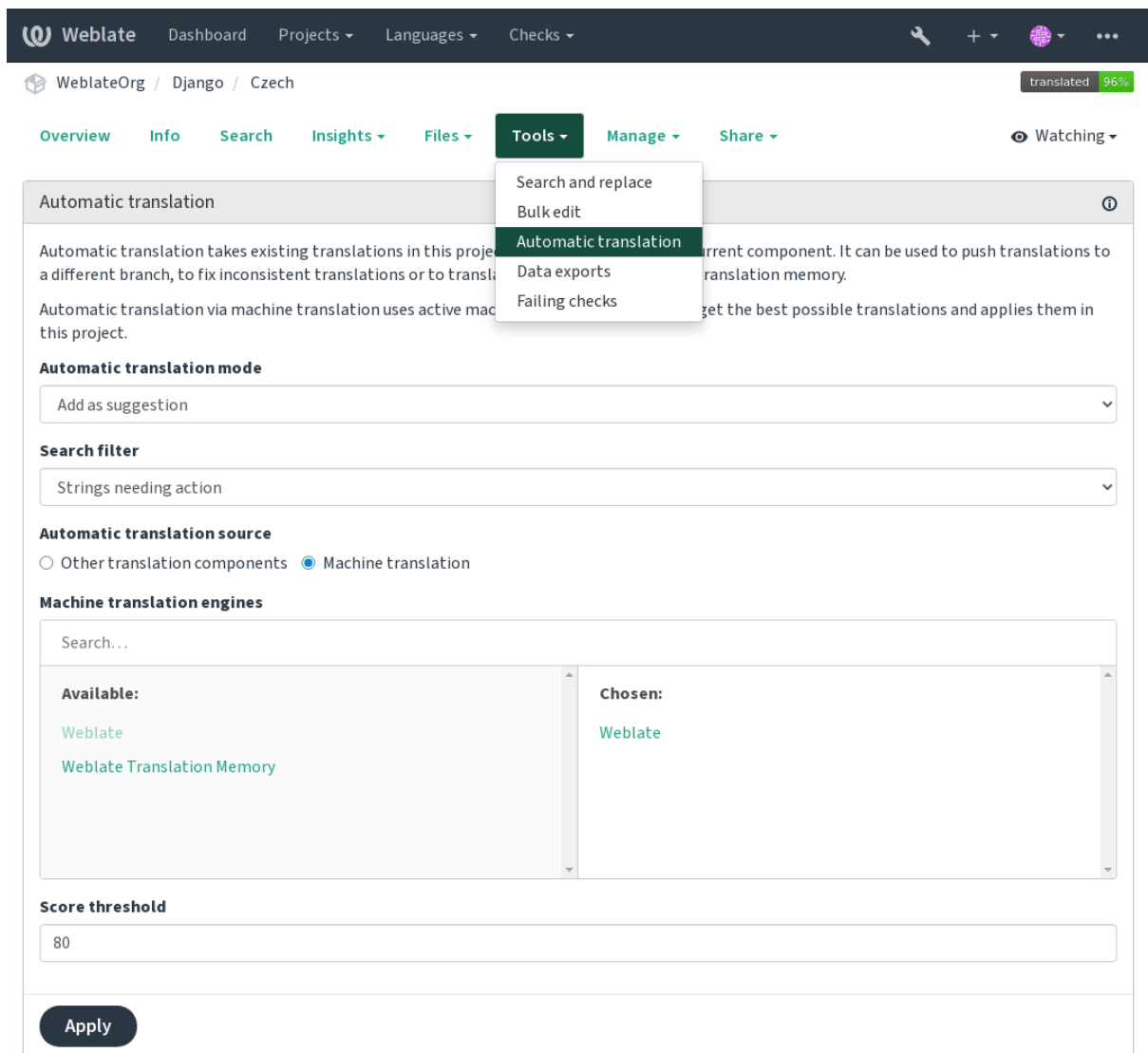
Based on configuration and your translated language, Weblate provides suggestions from several machine translation tools and [Memoria de traducción](#). All machine translations are available in a single tab of each translation page.

Ver también:

You can find the list of supported tools in [Traducción automática](#).

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:



Powered by [Weblate 4.6](#) [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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).

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: Don't worry about messing up the strings. This is a two-step process showing a preview of edited strings before the actual change is confirmed.

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 addon](#)

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:

- PO de gettext
- XLIFF con extensiones gettext
- XLIFF 1.1

- TermBase eXchange
- Translation Memory eXchange
- MO de gettext
- CSV
- Open XML de Excel
- JSON
- Recurso de cadenas de Android
- Cadenas de iOS

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The 'Files' dropdown menu is open, showing options: 'Download translation', 'Customize download' (highlighted), and 'Upload translation'. Below the menu, the 'Quick downloads' section lists three download options with their respective file formats and counts. The 'Customize download' section allows users to select the file format and download the translation.

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

Customize download

All strings ▾

File format

☒ gettext PO
 ☐ XLIFF with gettext extensions
 ☐ XLIFF 1.1
 ☐ TBX
 ☐ TMX
 ☐ gettext MO
 ☐ CSV
 ☐ XLSX
 ☐ JSON
 ☐ Android String Resource
 ☐ iOS strings

Download

Ver también:

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

1.4.2 Cargar traducciones

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

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The 'Files' menu is open, showing options: 'Download translation', 'Customize download', and 'Upload translation'. The 'Upload translation' option is highlighted. Below the menu, the 'Upload' dialog box is displayed. It contains the following fields and options:

- File**: A 'Choose File' button and the text 'No file chosen'.
- File upload mode**: Radio buttons for 'Add as translation', 'Add as suggestion', 'Add as translation needing edit', and 'Replace existing translation file'.
- Processing of strings needing edit**: A dropdown menu set to 'Do not import'.
- Conflict handling**: A dropdown menu set to 'Update translated strings'.
- Author name**: A text input field containing 'Weblate Test'.
- Author e-mail**: A text input field containing 'weblate@example.org'.
- Upload**: A dark blue button at the bottom.

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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

El archivo cargado se combina con el del repositorio para actualizar la traducción, de manera predeterminada sobrescribiendo las entradas existentes (en el diálogo de carga se puede activar o desactivar este comportamiento).

Métodos de importación

Estas son las elecciones presentadas al cargar archivos de traducción:

Añadir como traducción (translate**)** Las traducciones importadas se añaden como traducciones adicionales. Este constituye el caso de uso más común y es el comportamiento predeterminado.

Añadir como sugerencia (suggest**)** Las traducciones importadas se añaden como sugerencias. Recomendado si quiere que otros revisen lo que se ha cargado.

Añadir como traducción que requiere edición (fuzzy**)** Imported translations are added as translations needing edit. This can be useful when you want translations to be used, but also reviewed.

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*.

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.

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:

Webblate Dashboard Projects Languages Checks

WebblateOrg / Glossary WebblateOrg / Czech translated 100%

Overview Info Search Insights Files Tools Share Not watching

Translation status

2 Strings 100%

3 Words 100%

Add new glossary term Browse Translate

Strings status

2 All strings — 3 words Browse Edit Zen

2 Translated strings — 3 words Browse Edit Zen

Other components

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

Browse all components

Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate

Puede explorar todos los términos del glosario:

Webblate Dashboard Projects Languages Checks

WebblateOrg / Glossary WebblateOrg / Czech / Browse translated 100%

All strings Source string Add new glossary term

English	Czech
machine translation	strojový překlad
project	projekt

Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate

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.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb trail reads 'WeblateOrg / Glossary / WeblateOrg / Czech / Translate'. A status bar indicates 'translated 100%'. The main area is titled 'Glossary term' and shows the English string 'project' and its Czech translation 'projekt'. There are buttons for 'Save', 'Suggest', 'Skip', and a 'Tools' dropdown menu. The 'Tools' menu is open, showing options: 'Delete string', 'Mark as read-only', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. On the right, a sidebar contains a 'Glossary' section with a search bar and a list of terms, and a 'String information' section showing 'String age' and 'Source string age' as '5 seconds ago'. At the bottom, there's a footer with 'Powered by Weblate 4.6' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Términos no traducibles

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 read-only* 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 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
 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

Czech, Few

několik slov

Czech, Other

%(count)s slov

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

☐ Needs editing

Save

Suggest

Skip

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format 1

Following format strings are missing: %(count)s

Dismiss

☐ For all languages

Missing plurals 2

Some plural forms are not translated

Dismiss

☐ For all languages

Glossary

English Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

21 seconds ago

Source string age

22 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGE S/django.po, string 5

pending

Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate

26

Capítulo 1. Documentación para usuarios

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

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

BBcode in translation does not match source

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

Text contains the same word twice in a row:

Nuevo en la versión 4.1.

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.

La traducción no sigue los términos definidos en un glosario.

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.

Ver también:

Glosario, Personalizar el comportamiento mediante indicadores, Indicadores de traducción

Espacio duplicado

Translation contains 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

Comprueba que el formato en las cadenas esté reproducido tal cual tanto en el origen como en la traducción. Omitir el formato en las traducciones por lo general provoca problemas graves; por esta razón el formato de las cadenas casi siempre debe coincidir con el origen.

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

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

Consejo: In case specific format check is not available in Weblate, you can use generic [Sustitutorios](#).

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

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

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

translated 96%

1 / 1

Custom search

Position and priority

Zen

Translation

English

Singular

% (count)s word

Plural

% (count)s words

Czech, One

% (count)s slovo

Czech, Few

% (count)s slova

Czech, Other

% (count)s slov

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

☐ Needs editing

Save

Suggest

Skip

Nearby strings 20
Comments
Automatic suggestions
Other languages 3

History

No matching activity found.

Browse all component changes

Glossary

EnglishCzech

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
16 seconds ago

Source string age
16 seconds ago

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

Powered by Weblate 4.6

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

Cadena de interpolación de AngularJS

AngularJS interpolation strings do not match source

Named format string	Su saldo es de {{amount}} {{ currency }}
Indicador que activar	<i>angularjs-format</i>

Ver también:

Interpolación de texto de AngularJS

Formato C

C format string does not match source

Simple format string	Hay %d manzanas
Position format string	Su saldo es de %1\$d %2\$s
Indicador que activar	<i>c-format</i>

Ver también:

[C format strings](#), [C printf format](#)

Formato C#

C# format string does not match source

Position format string	Hay {0} manzanas
Indicador que activar	<i>c-sharp-format</i>

Ver también:

[C# String Format](#)

Literales de plantilla ECMAScript

ECMAScript template literals do not match source

Interpolación	Hay \${number} manzanas
Indicador que activar	<i>es-format</i>

Ver también:

[Template literals](#)

Interpolación con i18next

The i18next interpolation does not match source

Nuevo en la versión 4.0.

Interpolación	There are {{number}} apples
Nesting	There are \$t(number) apples
Indicador que activar	<i>i18next-interpolation</i>

Ver también:

[i18next interpolation](#)

Formato Java

Java format string does not match source

Simple format string	Hay %d manzanas
Position format string	Su saldo es de %1\$d %2\$s
Indicador que activar	<i>java-format</i>

Ver también:

[Java Format Strings](#)

MessageFormat de Java

Java MessageFormat string does not match source

Position format string	Hay {0} manzanas
Indicador que activar	<i>java-messageformat</i> enables the check unconditionally
	<i>auto-java-messageformat</i> enables check only if there is a format string in the source

Ver también:

[Java MessageFormat](#)

Formato JavaScript

JavaScript format string does not match source

Simple format string	Hay %d manzanas
Indicador que activar	<i>javascript-format</i>

Ver también:

[JavaScript formatting strings](#)

Formato Lua

Lua format string does not match source

Simple format string	Hay %d manzanas
Indicador que activar	<i>lua-format</i>

Ver también:

[Lua formatting strings](#)

Sustitutorios con signo de porcentaje

The percent placeholders do not match source

Nuevo en la versión 4.0.

Simple format string	There are %number% apples
Indicador que activar	<i>percent-placeholders</i>

Formato Perl

Perl format string does not match source

Simple format string	Hay %d manzanas
Position format string	Su saldo es de %1\$d %2\$s
Indicador que activar	<i>perl-format</i>

Ver también:

[Perl sprintf](#), [Perl Format Strings](#)

Formato PHP

PHP format string does not match source

Simple format string	Hay %d manzanas
Position format string	Su saldo es de %1\$d %2\$s
Indicador que activar	<i>php-format</i>

Ver también:

[PHP sprintf documentation](#), [PHP Format Strings](#)

Formato de llaves de Python

Python brace format string does not match source

Simple format string	There are {} apples
Named format string	Your balance is {amount} {currency}
Indicador que activar	<i>python-brace-format</i>

Ver también:

[Python brace format](#), [Python Format Strings](#)

Formato Python

Python format string does not match source

Simple format string	Hay %d manzanas
Named format string	Your balance is %(amount) %(currency)
Indicador que activar	<i>python-format</i>

Ver también:

[Python string formatting](#), [Python Format Strings](#)

Formato Qt

Qt format string does not match source

Position format string	There are %1 apples
Indicador que activar	<i>qt-format</i>

Ver también:

[Qt QString::arg\(\)](#)

Formato de plurales de Qt

Qt plural format string does not match source

Plural format string	There are %Ln apple(s)
Indicador que activar	<i>qt-plural-format</i>

Ver también:

[Qt i18n guide](#)

Formato Ruby

Ruby format string does not match source

Simple format string	Hay %d manzanas
Position format string	Your balance is %1\$f %2\$s
Named format string	Your balance is %+.2<amount>f %<currency>s
Named template string	Your balance is %{amount} %{currency}
Indicador que activar	<i>ruby-format</i>

Ver también:

[Ruby Kernel#sprintf](#)

Formato Scheme

Scheme format string does not match source

Simple format string	Hay ~d manzanas
Indicador que activar	<i>scheme-format</i>

Ver también:

[Srfi 28](#), [Chicken Scheme format](#), [Guile Scheme formatted output](#)

Formato de Vue I18n

The Vue I18n formatting does not match source

Named formatting	There are {count} apples
Rails i18n formatting	There are %{count} apples
Linked locale messages	@:message.dio @:message.the_world!
Indicador que activar	<i>vue-format</i>

Ver también:

[Vue I18n Formatting](#), [Vue I18n Linked locale messages](#)

Se había traducido

This string has been translated in the past

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

Incoherente

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

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.

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 not translated 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](#) addon 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

The decorative kashida letters should not be used

Nuevo en la versión 3.5.

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

Ver también:

[Kashida on Wikipedia](#)

Enlaces de Markdown

Markdown links do not match source

Nuevo en la versión 3.5.

Markdown links do not match source.

Ver también:

[Enlaces de Markdown](#)

Referencias de Markdown

Markdown link references do not match source

Nuevo en la versión 3.5.

Markdown link references do not match source.

Ver también:

[Markdown links](#)

Sintaxis de Markdown

Markdown syntax does not match source

Nuevo en la versión 3.5.

La sintaxis de Markdown no coincide con la de origen

Ver también:

[Markdown span elements](#)

Longitud máxima de la traducción

Translation should not exceed given 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.

Tamaño máximo de la traducción

Translation rendered text should not exceed given 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.

Ver también:

[Gestionar tipos de letra](#), [Personalizar el comportamiento mediante indicadores](#), [Longitud máxima de la traducción](#)

\n desiguales

Number of \n in translation does not match source

Por lo general, los saltos de renglón codificados con escape son importantes para dar formato a la salida de un programa. La comprobación emite un error si la cantidad de literales `\n` en la traducción difiere de la cantidad en la cadena de origen.

Dos puntos desiguales

Source and translation do not both end with a 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

Source and translation do not both end with an 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

Source and translation do not both end with an exclamation mark

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

Source and translation do not both end with a full 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

Source and translation do not both end with a question mark

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

Source and translation do not both end with a semicolon

Comprueba que los punto y coma al final de las oraciones de la cadena de origen y de la traducción sean una réplica mutua. Esto puede resultar útil para garantizar un buen formato de cadenas tales como las de los archivos .desktop.

Ver también:

[Semicolon on Wikipedia](#)

Saltos de renglón desiguales

Number of new lines in translation does not match source

Por lo general, los saltos de renglón son importantes para dar formato a la salida de un programa. La comprobación emite un error si la cantidad de literales `\n` en la traducción difiere de la cantidad en la cadena de origen.

Faltan plurales

Some plural forms are not translated

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

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

Sustitutorios

Translation is missing some placeholders:

Nuevo en la versión 3.9.

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

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

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

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

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

Ver también:

Personalizar el comportamiento mediante indicadores

Espaciado de puntuación

Missing non breakable space before double punctuation sign

Nuevo en la versión 3.9.

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

Translation does not match regular expression:

Nuevo en la versión 3.9.

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

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

Mismos plurales

Some plural forms are translated in the same way

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

Source and translation do not both start with a 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

Source and translation do not both start with same number of spaces

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

Source and translation do not both end with a 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

Source and translation do not both end with a 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

Source and translation are identical

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.

Ver también:

Configuración de componentes, Personalizar el comportamiento mediante indicadores

HTML inseguro

The translation uses unsafe HTML markup

Nuevo en la versión 3.9.

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.

Ver también:

The HTML check is performed by the [Bleach](#) library developed by Mozilla.

URL

The translation does not contain an URL

Nuevo en la versión 3.5.

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

XML tags in translation do not match source

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.

Sintaxis XML

The translation is not valid XML

Nuevo en la versión 2.8.

The XML markup is not valid.

Espacio de anchura cero

Translation contains extra zero-width space character

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

The string uses three dots (...) instead of an ellipsis character (...)

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](#)

Largamente no traducida

The string has not been translated for a long time

Nuevo en la versión 4.1.

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

Varias comprobaciones fallidas

The translations in several languages have failing checks

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

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

Nuevo en la versión 4.1.

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

The string is used as plural, but not using plural forms

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 gettext

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

1.7 Búsquedas

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.

When no field is defined, the lookup happens on *Source*, *Translate* and *Context* fields.

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

- Review strings changed by other users**: `changed:>=2021-03-18 AND NOT changed_by:testuser`
- Translated strings**: `state:>=translated`
- Strings with comments**: `has:comment`
- Strings with any failing checks**: `has:check`
- Strings with suggestions from others**: `has:suggestion AND NOT suggestion_author:testuser`
- Approved strings with suggestions**: `state:approved AND has:suggestion`
- All untranslated strings added the past month**: `added:>=2021-03-18 AND state:<=needs-editing`
- Translated strings in a certain language**: `is:translated AND language:cs`

At the bottom of the search area is a 'Search' button. Below the search interface, there's a footer with 'Powered by Weblate 4.6' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

1.7.1 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'.

1.7.2 Campos

source:TEXT Source string case insensitive search.

target:TEXT Target string case insensitive search.

context:TEXT Context string case insensitive search.

key:TEXT Key string case insensitive search.

note:TEXT Comment string case insensitive search.

location:TEXT Location string case insensitive search.

priority:NÚMERO String priority.

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

state:TEXT State search (approved, translated, needs-editing, empty, read-only), supports *Operadores de campo*.

pending:BOOLEANO String pending for flushing to VCS.

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

is:TEXT Search for string states (pending, translated, untranslated).

language:TEXT String target language.

component:TEXT Component slug, see «*Slug*» del componente.

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

changed_by:TEXT 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:TEXT String has failing check.

dismissed_check:TEXT String has dismissed check.

comment:TEXT Search in user comments.

comment_author:TEXT Filter by comment author.

suggestion:TEXT Search in suggestions.

suggestion_author:TEXT Filter by suggestion author.

explanation:TEXT Buscar en las explicaciones.

1.7.3 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)`

1.7.4 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.

1.7.5 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`.

1.7.6 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"`.

1.7.7 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]"`.

1.7.8 Consultas predefinidas

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

1.7. Búsquedas

1.7.9 Ordenar los resultados

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

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. The breadcrumb trail is 'WeblateOrg / Django / Czech / Translate'. A status bar at the top right indicates 'translated 96%'. Below the navigation, there are controls for navigating between strings (1/1), a filter for 'Not translated strings', and a 'Zen' view toggle. The main area displays a 'Translation' card for an English string: 'The string uses three dots (...) instead of an ellipsis character (...)'. The Czech translation field is empty. A dropdown menu is open, showing sorting options: 'Position and priority', 'Position', 'Priority', 'Labels', 'Source string', 'Translated string', 'Age of string', 'Number of words', 'Number of comments', 'Number of failing checks', and 'Key'. Below the translation card, there are tabs for 'Nearby strings' (16), 'Comments', 'Automatic suggestions', and 'Other languages' (3). A 'History' section is also visible. On the right, a sidebar provides additional information: 'Explanation' (No explanation currently provided), 'Labels' (No labels currently set), 'Flags' (No flags currently set), 'Source string location' (weblate/checks/source.py:54), 'String age' (18 seconds ago), 'Source string age' (18 seconds ago), and 'Translation file' (weblate/locale/cs/LC_MESSAGE S/django.po, string 26). At the bottom, there is a 'New comment' section with a text area and a 'Save' button.

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 Translation states

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.

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»* addon 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	<i>Users</i>	Or <i>Translate</i> with <i>per-project access control</i> .
Grupo de revisores	Sin datos	Not used.

1.8.4 Revisión por pares

With this workflow, anybody can add suggestions, and need approval from additional member(s) before it is accepted as a translation.

- *Any user* can add suggestions.
- *Any user* can vote for suggestions.
- Suggestions become translations when given a predetermined number of votes.

Configuración	Value	Nota
Activar revisiones	desactivada	Configured at project level.
Activar sugerencias	activada	
Votar sugerencias	desactivada	
Aceptar sugerencias automáticamente	1	You can set higher value to require more peer reviews.
Grupo de traductores	<i>Users</i>	Or <i>Translate</i> with <i>per-project access control</i> .
Grupo de revisores	Sin datos	Not used, all translators review.

1.8.5 Revisores dedicados

Nuevo en la versión 2.18: The proper review workflow is supported since Weblate 2.18.

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	<i>Users</i>	Or <i>Translate</i> with <i>per-project access control</i> .
Grupo de revisores	<i>Revisores</i>	Or <i>Review</i> with <i>per-project access control</i> .

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):

Webate Dashboard Projects Languages Checks

WebateOrg / Settings

Basic Access **Workflow** Components

☒ **Set "Language-Team" header**
Lets Weblate update the "Language-Team" file header of your project.

☒ **Use shared translation memory**
Uses the pool of shared translations between projects.

☒ **Contribute to shared translation memory**
Contributes to the pool of shared translations between projects.

☒ **Enable hooks**
Whether to allow updating this repository by remote hooks.

Language aliases ⓘ

Comma-separated list of language code mappings, for example: en_GB:en,en_US:en

☐ **Enable reviews**
Requires dedicated reviewers to approve translations.

☐ **Enable source reviews**
Requires dedicated reviewers to approve source strings.

Save

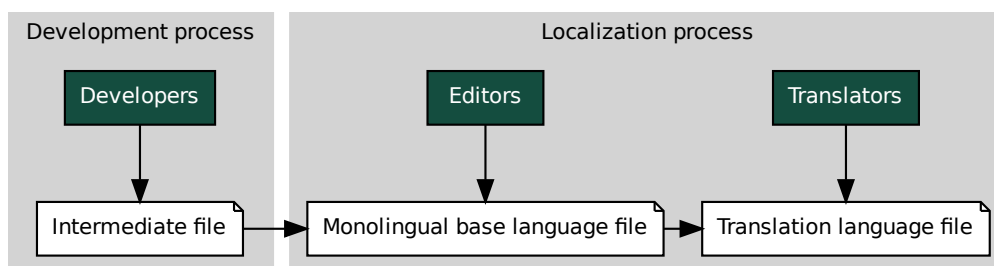
Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate

Nota: Depending on Weblate configuration, the setting might not be available to you. For example on Hosted Weblate this is not available for projects hosted for free.

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

With *Activar revisiones de origen* enabled, the review process can be applied on the source strings. Once enabled, users can report issues in the source strings. The actual process depends on whether you use bilingual or monolingual formats.

For monolingual formats, the source string review behaves similarly as with *Revisores dedicados* - once issue is reported on the source string, it is marked as *Needs editing*.

The bilingual formats do not allow direct editing of the source strings (these are typically extracted directly from the source code). In this case *Source needs review* label is attached to strings reported by translators. You should review such strings and either edit them in the source or remove the label.

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 ran into a merge conflict, the easiest way is to solve all conflicts locally at your workstation - is to simply 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

# 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
... # Resolve conflicts
git commit
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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.

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

This is the default committer name, configured when you create a translation component. You can change it in the administration at any time.

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

Ver también:

Configuración de componentes

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:

Traducción automática, 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:

Language definitions

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.

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

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

In case you want to do the update automatically, you can install addon *Actualizar archivos PO para que coincidan con POT (msgmerge)*.

Ver también:

updating-target-files

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.
- 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 [Parsing language codes](#) 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 *GitHub*, *Gerrit* and *Subversion*) 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:

`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 4646](#) 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_Hant`.

Ver también:

Language definitions

1.10 Formatos de archivo admitidos

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

Ver también:

[Translation Related File Formats](#)

Nota: 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 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.

1.10.2 Detección automática

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

1.10.3 Translation types capabilities

Capacidades de todos los formatos admitidos:

Formato	Lingua- lity ^{Página 60, 1}	Plura- les ^{Página 60, 2}	Comenta- rios ^{Página 60, 3}	Contex- to ^{Página 60, 4}	Ubica- ción ^{Página 60, 5}	Flags ^{Página 60, 8}	Estados adiciona- les ^{Página 60, 6}
<i>gettext de GNU</i>	bilingual	yes	yes	yes	yes	yes ⁹	needs edi- ting
<i>Mono- lingual gettext</i>	mono	yes	yes	yes	yes	yes ⁷	needs edi- ting
<i>XLIFF</i>	both	yes	yes	yes	yes	yes ¹⁰	needs editing, approved
<i>Java properties</i>	both	no	yes	no	no	no	
<i>Propie- dades GWT</i>	mono	yes	yes	no	no	no	

continué en la próxima página

Tabla 1 – proviene de la página anterior

Formato	Lingua- lity ^{Página 60, 1}	Plura- les ^{Página 60, 2}	Comenta- rios ^{Página 60, 3}	Contex- to ^{Página 60, 4}	Ubica- ción ^{Página 60, 5}	Flags ^{Página 60, 8}	Estados adiciona- les ^{Página 60, 6}
<i>Traduccio- nes para Joomla</i>	mono	no	yes	no	yes	no	
<i>.ts de Qt Linguist</i>	both	yes	yes	no	yes	yes [?]	needs edi- ting
<i>Recursos de cadenas de Android</i>	mono	yes	yes [?]	no	no	yes [?]	
<i>Cadenas de iOS de Ap- ple</i>	bilingual	no	yes	no	no	no	
<i>Cadenas de PHP</i>	mono	no ¹¹	yes	no	no	no	
<i>Archivos JSON</i>	mono	no	no	no	no	no	
<i>Archivos JSON de i18next</i>	mono	yes	no	no	no	no	
<i>Archivos JSON de go-i18n</i>	mono	yes	no	no	no	no	
<i>Archivo ARB</i>	mono	yes	yes	no	no	no	
<i>JSON para WebExten- sion</i>	mono	yes	yes	no	no	no	
<i>.XML resource files</i>	mono	no	yes	no	no	yes [?]	
<i>Archivos CSV</i>	both	no	yes	yes	yes	no	needs edi- ting
<i>YAML files</i>	mono	no	yes	no	no	no	
<i>Ruby YAML files</i>	mono	yes	yes	no	no	no	
<i>DTD files</i>	mono	no	no	no	no	no	
<i>Flat XML files</i>	mono	no	no	no	no	yes [?]	
<i>Archivos RC de Windows</i>	mono	no	yes	no	no	no	
<i>Open XML de Excel</i>	mono	no	yes	yes	yes	no	needs edi- ting
<i>Archivos de meta- datos de tiendas de aplicacio- nes</i>	mono	no	no	no	no	no	
<i>Archi- vos de subtítulos</i>	mono	no	no	no	yes	no	

continué en la próxima página

Tabla 1 – proviene de la página anterior

Formato	Lingua- lity ^{Página 60, 1}	Plura- les ^{Página 60, 2}	Comenta- rios ^{Página 60, 3}	Contex- to ^{Página 60, 4}	Ubica- ción ^{Página 60, 5}	Flags ^{Página 60, 8}	Estados adiciona- les ^{Página 60, 6}
<i>Archivos HTML</i>	mono	no	no	no	no	no	
<i>Formato OpenDocument</i>	mono	no	no	no	no	no	
<i>Formato IDML</i>	mono	no	no	no	no	no	
<i>Traducciones en INI</i>	mono	no	no	no	no	no	
<i>Traducciones INI de Inno Setup</i>	mono	no	no	no	no	no	
<i>TermBase eXchange format</i>	bilingual	no	yes	no	no	yes ⁷	
<i>Archivos de texto</i>	mono	no	no	no	no	no	

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*.

1.10.4 gettext de GNU

El formato más ampliamente utilizado para traducir software libre.

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

The bilingual gettext PO file typically looks like this:

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

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

(continúe en la próxima página)

¹ Vea *Formatos bilingües y monolingües*

² Los plurales son necesarios para regionalizar adecuadamente las cadenas que tienen recuento variable.

³ Los comentarios pueden emplearse para compartir información adicional sobre la cadena que se va a traducir.

⁴ El contexto se utiliza para discernir cadenas idénticas que se utilizan en ámbitos distintos (por ejemplo, *Sun* puede emplearse como nombre abreviado del día «Sunday» o como el nombre de la estrella más cercana a nosotros).

⁵ La ubicación en código fuente de una cadena puede ayudar a los traductores avezados a entender de qué manera una determinada cadena se utilizará.

⁸ See *Personalizar el comportamiento mediante indicadores*

⁶ Additional states supported by the file format in addition to «Not translated» and «Translated».

⁹ The gettext type comments are used as flags.

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

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

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

(proviene de la página anterior)

```
msgstr "Úterý"

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

Typical Weblate <i>Configuración de componentes</i>	
Máscara de archivos	po/* .po
Archivo de base monolingüe	<i>Empty</i>
Plantilla para traducciones nuevas	po/messages.pot
Formato de archivo	<i>Gettext PO file</i>

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)*

Monolingual gettext

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

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"
```

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

1.10.5 XLIFF

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

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

Ver también:

XML Localization Interchange File Format (XLIFF) specification

Translation states

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).

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

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

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

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

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>
```

Specifying translation flags

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*).

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.

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

Typical Weblate <i>Configuración de componentes</i> for monolingual XLIFF	
File mask	<code>localizations/*.xliff</code>
Archivo de base monolingüe	<code>localizations/en-US.xliff</code>
Plantilla para traducciones nuevas	<code>localizations/en-US.xliff</code>
Formato de archivo	<i>XLIFF Translation File</i>

Ver también:

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

1.10.6 Java properties

Native Java format for translations.

Java properties are usually used as monolingual translations.

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

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.

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

Ver también:

[Java properties on Wikipedia](#), [Mozilla and Java properties files](#), *Formatea el archivo de propiedades de Java*, *Limpieza de archivos de traducción*

1.10.7 Propiedades GWT

Native GWT format for translations.

GWT properties are usually used as monolingual translations.

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

Ver también:

[GWT localization guide](#), [GWT Internationalization Tutorial](#), [Mozilla and Java properties files](#), *Formatea el archivo de propiedades de Java*, *Limpieza de archivos de traducción*

1.10.8 Traducciones en INI

Nuevo en la versión 4.1.

INI file format for translations.

INI translations are usually used as monolingual translations.

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

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 *Java properties* instead.

Ver también:

[INI Files](#), [Java properties](#), [Traducciones para Joomla](#), [Traducciones INI de Inno Setup](#)

1.10.9 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 sustitutorios `%n` y `%t` para insertar saltos de renglón y tabuladores, respectivamente.

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

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

Ver también:

[INI Files](#), [Traducciones para Joomla](#), [Traducciones en INI](#)

1.10.10 Traducciones para Joomla

Nuevo en la versión 2.12.

Formato nativo de Joomla para traducciones.

Joomla translations are usually used as monolingual translations.

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

Ver también:

Specification of Joomla language files, Mozilla and Java properties files, [Traducciones en INI](#), [Traducciones INI de Inno Setup](#)

1.10.11 .ts de Qt Linguist

Formato de traducción utilizado en aplicaciones programadas con Qt.

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

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

Typical Weblate <i>Configuración de componentes</i> 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	<i>Archivo de traducción de Qt Linguist</i>

Ver también:

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

1.10.12 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 others `res/values/strings.xml`.

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

Ver también:

Android string resources documentation, Android string resources

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>

1.10.13 Cadenas de iOS de Apple

Apple specific file format for translating applications, used for both iOS and iPhone/iPad application translations.

Apple iOS strings are usually used as bilingual translations.

Typical Weblate <i>Configuración de componentes</i>	
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	<i>Empty</i>
Formato de archivo	<i>Cadenas de iOS (UTF-8)</i>

Ver también:

Apple «strings files» documentation, Mac OSX strings

1.10.14 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.

Archivo de ejemplo:

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

Typical Weblate <i>Configuración de componentes</i>	
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	<i>Cadenas de PHP</i>

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

1.10.15 Archivos JSON

Nuevo en la versión 2.0.

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

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.
- *Archivos JSON de i18next*
- *Archivos JSON de go-i18n*
- *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.

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": ""
  }
}
```

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"
  }
}
```

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

Ver también:

JSON, *Personalizar la salida de JSON*, *Limpieza de archivos de traducción*,

1.10.16 Archivos JSON de i18next

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

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

Archivo de ejemplo:

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

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

Ver también:

JSON, i18next JSON Format, *Personalizar la salida de JSON*, *Limpieza de archivos de traducción*

1.10.17 Archivos JSON de go-i18n

Nuevo en la versión 4.1.

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 format, for flat JSON formats please use *Archivos JSON*. The v2 format with hash is currently not supported.

Typical Weblate <i>Configuración de componentes</i>	
Máscara de archivos	langs/*.json
Archivo de base monolingüe	langs/en.json
Plantilla para traducciones nuevas	<i>Empty</i>
Formato de archivo	<i>Archivo JSON de go-i18n</i>

Ver también:

[JSON](#), [go-i18n](#), [Personalizar la salida de JSON](#), [Limpieza de archivos de traducción](#),

1.10.18 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.

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

Ver también:

[JSON](#), [Application Resource Bundle Specification](#), [Internationalizing Flutter apps](#), [Personalizar la salida de JSON](#), [Limpieza de archivos de traducción](#)

1.10.19 JSON para WebExtension

Nuevo en la versión 2.16: This is supported since Weblate 2.16 and with [translate-toolkit](#) at-least 2.2.4.

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

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.

Archivo de ejemplo:

```
{
  "hello": {
    "message": "Ahoj světe!\n",
    "description": "Description",
    "placeholders": {
      "url": {
        "content": "$1",
        "example": "https://developer.mozilla.org"
      }
    }
  },
  "orangutan": {
    "message": "",
    "description": "Description"
  },
  "try": {
    "message": "",
```

(continué en la próxima página)

(proviene de la página anterior)

```

    "description": "Description"
  },
  "thanks": {
    "message": "",
    "description": "Description"
  }
}

```

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

Ver también:

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

1.10.20 .XML resource files

Nuevo en la versión 2.3.

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

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

Ver también:

.NET Resource files (.resx), [Limpieza de archivos de traducción](#)

1.10.21 Archivos CSV

Nuevo en la versión 2.4.

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

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

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.

Archivo de ejemplo:

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

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

Typical Weblate <i>Configuración de componentes</i> 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	<i>Simple CSV file</i>

Ver también:

[CSV](#)

1.10.22 YAML files

Nuevo en la versión 2.9.

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

Example of a YAML file:

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

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

Ver también:

[YAML](#), [Ruby YAML files](#)

1.10.23 Ruby YAML files

Nuevo en la versión 2.9.

Ruby i18n YAML files with language as root node.

Example Ruby i18n YAML file:

```
cs:
  weblate:
    hello: ""
    orangutan: ""
```

(continué en la próxima página)

(proviene de la página anterior)

```
try: ""
thanks: ""
```

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

Ver también:[YAML](#), [YAML files](#)

1.10.24 DTD files

Nuevo en la versión 2.18.

Example DTD file:

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

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

Ver también:[Mozilla DTD format](#)

1.10.25 Flat XML files

Nuevo en la versión 3.9.

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>
```

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

Ver también:[Flat XML](#)

1.10.26 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 currently in beta, feedback from testing is welcome.

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
```

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

Ver también:

[Windows RC files](#)

1.10.27 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.

Typical Weblate <i>Configuración de componentes</i>	
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	<i>Archivos de metadatos de tiendas de aplicaciones</i>

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](#).

1.10.28 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)

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

Ver también:

[Subtitles](#)

1.10.29 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.

1.10.30 Archivos HTML

Nuevo en la versión 4.1.

Nota: Support for this format is currently in beta, feedback from testing is welcome.

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

Ver también:

[HTML](#)

1.10.31 Archivos de texto

Nuevo en la versión 4.6.

Nota: Support for this format is currently in beta, feedback from testing is welcome.

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

There are three flavors of this format:

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

Ver también:

[Simple Text Documents](#)

1.10.32 Formato OpenDocument

Nuevo en la versión 4.1.

Nota: Support for this format is currently in beta, feedback from testing is welcome.

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

Ver también:

[OpenDocument Format](#)

1.10.33 Formato IDML

Nuevo en la versión 4.1.

Nota: Support for this format is currently in beta, feedback from testing is welcome.

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

1.10.34 TermBase eXchange format

Nuevo en la versión 4.5.

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

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

Ver también:

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

1.10.35 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 [GitHub](#), [Gerrit](#) and [Subversion](#)) 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:

The screenshot shows the Weblate admin interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a sub-header 'Manage / SSH keys'. A horizontal menu contains links to Weblate status, Backups, Translation memory, Performance report, SSH keys (highlighted), Alerts, Repositories, Users, and Appearance. Below the menu, the 'Public SSH key' section shows the current SSH key used by Weblate, with a 'Download private key' button. The 'Known host keys' section displays a table of host keys. The 'Add host key' section provides a form to add new host keys.

Public SSH key

Weblate currently uses this SSH key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCDkUMKvxqBGmPWC0Tnn7az4C/VTyPhqCr8FxBfVfVGHvSxcFi3VWmGvsq3+sVXvqPoiG9nEU11StMje8KBwNGlghtvHM
```

[Download private key](#)

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname **Port**

[Submit](#)

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Clave SSH de Weblate

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:

The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a sub-header 'Manage / SSH keys'. A yellow notification banner states: 'Added host key for github.com with fingerprint nThbg6kXUpJWG17E1GOCspRomTxdCARLviKw6E5SY8 (ssh-rsa), please verify that it is correct.' The main navigation bar includes links for Weblate status, Backups, Translation memory, Performance report, SSH keys (highlighted), Alerts, Repositories, Users, and Appearance. Below this is a secondary bar with Tools and Billing. The 'Public SSH key' section shows the current key type as 'ssh-rsa' and a long fingerprint string, with a 'Download private key' button. The 'Known host keys' section contains a table with columns for Hostname, Key type, and Fingerprint, listing github.com with ssh-rsa key type and fingerprint nThbg6kXUpJWG17E1GOCspRomTxdCARLviKw6E5SY8. The 'Add host key' section provides instructions on how to verify host keys and includes input fields for Hostname (pre-filled with github.com) and Port, with a Submit button. The footer contains the text 'Powered by Weblate 4.6' and links to About Weblate, Legal, Contact, Documentation, and Donate to Weblate.

Public SSH key

Weblate currently uses this SSH key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCDKumKvxqBGmPWC0Tnn7az4C/VTyPhqICr8FxRobfhVGHvufSxcFi3VvmGvsq3+sVXvqPoiiG9nEU11StMje8KBwNGightvHM
```

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWG17E1GOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname Port

Submit

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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 addons 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

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 GitHub

Nuevo en la versión 2.3.

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

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

Ver también:

[Enviar cambios efectuados en Weblate](#)

Enviar cambios a GitHub como solicitudes de incorporación

Si no se desea enviar las traducciones directamente a un repositorio en GitHub, estas pueden enviarse como una o múltiples solicitudes de incorporación.

You need to configure API credentials to make this work.

Ver también:

GITHUB_USERNAME, GITHUB_TOKEN, GITHUB_CREDENTIALS

1.11.4 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 *GitLab* creates merge request.

Ver también:

Enviar cambios efectuados en Weblate

Enviar cambios a GitLab como solicitudes de fusión

Si no se desea enviar las traducciones directamente a un repositorio en GitLab, estas pueden enviarse como una o múltiples solicitudes de fusión.

You need to configure API credentials to make this work.

Ver también:

GITLAB_USERNAME, GITLAB_TOKEN, GITLAB_CREDENTIALS

1.11.5 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 *Pagure* creates merge request.

Ver también:

Enviar cambios efectuados en Weblate

Enviar cambios a Pagure como solicitudes de fusión

Si no se desea enviar las traducciones directamente a un repositorio en Pagure, estas pueden enviarse como una o múltiples solicitudes de fusión.

You need to configure API credentials to make this work.

Ver también:

PAGURE_USERNAME, PAGURE_TOKEN, PAGURE_CREDENTIALS

1.11.6 Gerrit

Nuevo en la versión 2.2.

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

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

1.11.7 Mercurial

Nuevo en la versión 2.1.

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.

1.11.8 Subversion

Nuevo en la versión 2.8.

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

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.

Distinto en la versión 2.19: Before this, only repositories using the standard layout were supported.

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`

1.11.9 Archivos locales

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

Nuevo en la versión 2.6: La API REST está disponible a partir de Weblate 2.6.

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.

Query Parameters

- **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.

Request Headers

- `Accept` – the response content type depends on `Accept` header
- `Authorization` – optional token to authenticate

Response Headers

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

Response JSON Object

- **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

Status Codes

- `200 OK` – when request was correctly handled

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

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/
```

API rate limiting

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.

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

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

{
  "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)

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

Returns information about users.

Parámetros

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

Response JSON Object

- **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
- **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",
  "groups": [
    "http://example.com/api/groups/2/",
    "http://example.com/api/groups/3/"
  ],
  "is_superuser": true,
  "is_active": true,
  "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

Response JSON Object

- **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
- **date_joined** (*string*) – date the user is created

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

Changes the user parameters.

Parámetros

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

Response JSON Object

- **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
- **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

Form Parameters

- **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

Response JSON Object

- **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

Request JSON Object

- **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

Request JSON Object

- **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

Request JSON Object

- **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

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

Returns information about group.

Parámetros

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

Response JSON Object

- **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)/`
- **componentlist** (*array*) – link to associated componentlist; see `GET /api/component-lists/(str:slug)/`

Example JSON data:

```
{
  "name": "Guests",
  "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

Response JSON Object

- **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

Response JSON Object

- **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

Form Parameters

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

POST `/api/groups/(int: id)/components/`
Associate components with a group.

Parámetros

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

Form Parameters

- **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

Form Parameters

- **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

Form Parameters

- **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

Form Parameters

- **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

Response JSON Object

- **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

Response JSON Object

- **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

Response JSON Object

- **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
- **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

Response JSON Object

- **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

Example JSON data:

```
{
  "code": "en",
  "direction": "ltr",
  "name": "English",
  "plural": {
    "id": 75,
    "source": 0,
    "number": 2,
    "formula": "n != 1",
    "type": 1
  },
  "aliases": [
    "english",
    "en_en",
    "base",
    "source",
```

(continué en la próxima página)

(proviene de la página anterior)

```

    "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

Request JSON Object

- **name** (*string*) – Nombre del idioma
- **direction** (*string*) – Dirección del texto
- **plural** (*object*) – Language plural details

PATCH `/api/languages/(string: language) /`
 Permite cambiar los parámetros del idioma.

Parámetros

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

Request JSON Object

- **name** (*string*) – Nombre del idioma
- **direction** (*string*) – Dirección del texto
- **plural** (*object*) – Language plural details

DELETE `/api/languages/(string: language) /`
 Deletes the language.

Parámetros

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

GET `/api/languages/(string: language) /statistics/`
 Devuelve estadísticas sobre un idioma.

Parámetros

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

Response JSON Object

- **total** (*int*) – número total de cadenas
- **total_words** (*int*) – número total de palabras
- **last_change** (*timestamp*) – últimos cambios en el idioma
- **recent_changes** (*int*) – número total de cambios
- **translated** (*int*) – número de cadenas traducidas
- **translated_percent** (*float*) – porcentaje de cadenas traducidas
- **translated_words** (*int*) – número de palabras traducidas
- **translated_words_percent** (*int*) – porcentaje de palabras traducidas
- **translated_chars** (*int*) – número de caracteres traducidos
- **translated_chars_percent** (*int*) – porcentaje de caracteres traducidos

- **total_chars** (*int*) – número de caracteres totales
- **fuzzy** (*int*) – number of fuzzy (marked for edit) strings
- **fuzzy_percent** (*int*) – percentage of fuzzy (marked for edit) strings
- **failing** (*int*) – número de cadenas fallidas
- **failing** – porcentaje de cadenas fallidas

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

Response JSON Object

- **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

Response JSON Object

- **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

Response JSON Object

- **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

Request JSON Object

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

Response JSON Object

- **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+
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

Response JSON Object

- **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

Form Parameters

- **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](#).

Response JSON Object

- **result** (*object*) – Created component object; see [GET /api/components/\(string:project\)/\(string:component\)/](#)

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

CURL form request example:

```
curl \
  --form docfile=@strings.html \
  --form name=Weblate \
  --form slug=weblate \
  --form file_format=html \
  --form new_lang=add \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

CURL JSON request example:

```
curl \
  --data-binary '{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "slug": "weblate",
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
```

(continué en la próxima página)

(proviene de la página anterior)

```

    "new_base": "",
    "vcs": "git"
  }' \
-H "Content-Type: application/json" \
-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

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "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",
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    }
  },

```

(continué en la próxima página)

(proviene de la página anterior)

```

    "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

Response JSON Object

- **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

Response JSON Object

- **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

1.12.8 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

Response JSON Object

- **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*) – *File mask*
- **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*) – Mensajes de consigna, de adición, de eliminación, de fusión y de complemento
- **add_message** (*string*) – Mensajes de consigna, de adición, de eliminación, de fusión y de complemento
- **delete_message** (*string*) – Mensajes de consigna, de adición, de eliminación, de fusión y de complemento
- **merge_message** (*string*) – Mensajes de consigna, de adición, de eliminación, de fusión y de complemento
- **addon_message** (*string*) – Mensajes de consigna, de adición, de eliminación, de fusión y de complemento
- **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
- **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 to a background task (if any); see `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",
```

(continué en la próxima página)

(proviene de la página anterior)

```

    "source_language": {
      "code": "en",
      "direction": "ltr",
      "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",
    "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)

Request JSON Object

- **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",
      "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

Request JSON Object

- **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

Response JSON Object

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

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

Response JSON Object

- **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

Response JSON Object

- **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

Request JSON Object

- **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

Response JSON Object

- **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

Request JSON Object

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

Response JSON Object

- **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

Response JSON Object

- **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

Request JSON Object

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

Response JSON Object

- **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,
  "have_suggestion": 0,
  "is_template": false,
  "is_source": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
```

(continué en la próxima página)

(proviene de la página anterior)

```

    "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.

Nuevo en la versión 2.7.

Parámetros

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

Response JSON Object

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

GET `/api/components/(string: project) /`
string: `component/links/` Returns projects linked with a component.

Nuevo en la versión 4.5.

Parámetros

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

Response JSON Object

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

POST `/api/components/(string: project) /`
string: `component/links/` Associate project with a component.

Nuevo en la versión 4.5.

Parámetros

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

Form Parameters

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

DELETE `/api/components/(string: project) /`
string: `component/links/string: project_slug/` Remove association of a project with a component.

Nuevo en la versión 4.5.

Parámetros

- **project** (*string*) – URL semántico del proyecto
- **component** (*string*) – URL semántico del componente
- **project_slug** (*string*) – Slug of the project to remove

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

Response JSON Object

- **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*) – number of fuzzy (marked for edit) strings
- **fuzzy_percent** (*float*) – percentage of fuzzy (marked for edit) strings
- **fuzzy_words** (*int*) – number of words in fuzzy (marked for edit) strings
- **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",
        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
      },
      "url": "http://example.com/api/projects/hello/",
      "web": "https://weblate.org/",
      "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "slug": "weblate",
    "template": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
  },
  "failing_checks": 3,
  "failing_checks_percent": 75.0,
  "failing_checks_words": 11,
```

(continué en la próxima página)

(proviene de la página anterior)

```

"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",
  "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

Response JSON Object

- **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)

Response JSON Object

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

POST /api/translations/ (string: project) /
string: component/string: language/units/ Add new monolingual unit.

Parámetros

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

Request JSON Object

- **key** (*string*) – Nombre de la unidad de traducción
- **value** (*string*) – El valor de la unidad de traducción

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

Request JSON Object

- **mode** (*string*) – Modo de traducción automatizada
- **filter_type** (*string*) – Automatic translation filter type
- **auto_source** (*string*) – Fuente de la traducción automática
- **component** (*string*) – Permita que el proyecto contribuya a la memoria de traducción compartida para obtener acceso a componentes adicionales.
- **engines** (*string*) – 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.

Query Parameters

- **format** – File format to use; if not specified no format conversion happens; supported file formats: po, mo, xiff, xiff11, tbx, csv, xlsx, json, aresource, strings

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

Form Parameters

- **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

Request JSON Object

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

Response JSON Object

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

GET `/api/translations/ (string: project) /`
string: *component* / **string:** *language* / **statistics** / Devuelve estadísticas detalladas sobre la traducción.

Nuevo en la versión 2.7.

Parámetros

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

Response JSON Object

- **code** (*string*) – código del idioma
- **failing** (*int*) – número de comprobaciones fallidas
- **failing_percent** (*float*) – porcentaje de comprobaciones fallidas
- **fuzzy** (*int*) – number of fuzzy (marked for edit) strings
- **fuzzy_percent** (*float*) – percentage of fuzzy (marked for edit) strings
- **total_words** (*int*) – número total de palabras
- **translated_words** (*int*) – número de palabras traducidas
- **last_author** (*string*) – nombre del último autor
- **last_change** (*timestamp*) – fecha del último cambio
- **name** (*string*) – nombre del idioma
- **total** (*int*) – número total de cadenas
- **translated** (*int*) – número de cadenas traducidas
- **translated_percent** (*float*) – porcentaje de cadenas traducidas
- **url** (*string*) – URL to access the translation (engagement URL)
- **url_translate** (*string*) – URL to access the translation (real translation URL)

1.12.10 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.

Nuevo en la versión 2.10.

GET `/api/units/`
Devuelve un listado de unidades de traducción.

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

Response JSON Object

- **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
- **state** (*int*) – unit state, 0 - not translated, 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*) – Source unit link; see [GET /api/units/\(int:id\)/](#)

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

Nuevo en la versión 4.3.

Performs partial update on translation unit.

Parámetros

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

Request JSON Object

- **state** (*int*) – unit state, 0 - not translated, 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](#)

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

Nuevo en la versión 4.3.

Performs full update on translation unit.

Parámetros

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

Request JSON Object

- **state** (*int*) – unit state, 0 - not translated, 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](#)

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

Nuevo en la versión 4.3.

Deletes a translation unit.

Parámetros

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

1.12.11 Cambios

Nuevo en la versión 2.10.

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/\(int:id\)/](#).

Query Parameters

- **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

Response JSON Object

- **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

1.12.12 Capturas de pantalla

Nuevo en la versión 2.14.

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

Response JSON Object

- **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

Form Parameters

- **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

Form Parameters

- **string unit_id** – Identificador de unidad

Response JSON Object

- **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.

Form Parameters

- **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

Response JSON Object

- **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

Response JSON Object

- **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

Response JSON Object

- **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

1.12.13 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. de complemento

Response JSON Object

- **name** (*string*) – nombre de un complemento
- **component** (*string*) – URL of a related component object
- **configuration** (*object*) – Configuración opcional del complemento

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

Request JSON Object

- **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. de complemento

Response JSON Object

- **configuration** (*object*) – Configuración opcional del complemento

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

Editar información completa relativa a un complemento.

Parámetros

- **id** (*int*) – Id. de complemento

Response JSON Object

- **configuration** (*object*) – Configuración opcional del complemento

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

Eliminar el complemento.

Parámetros

- **id** (*int*) – Id. de complemento

1.12.14 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

Response JSON Object

- **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

Request JSON Object

- **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

Request JSON Object

- **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

Form Parameters

- **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

1.12.15 Glosario

Distinto en la versión 4.5: Glossaries are now stored as regular components, translations and strings, please use respective API instead.

1.12.16 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

Response JSON Object

- **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

1.12.17 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/github/extending-github/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/ce/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.

Special hook for handling Azure Repos notifications and automatically updating matching components.

Ver también:

[*Recibir cambios automáticamente de Azure Repos*](#) For instruction on setting up Azure integration

<https://docs.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops> Información genérica sobre los actuadores web de Azure Repos

[*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

1.12.18 Exportaciones

Weblate brinda diversas exportaciones para permitirle un tratamiento posterior de sus datos.

```
GET /exports/stats/(string: project) /
string: component/
```

Query Parameters

- **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.

Ejemplo de solicitud:

```
GET /exports/stats/weblate/main/ HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Ejemplo de respuesta:

```
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/"
    }
  ]

```

1.12.19 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 en Wikipedia](#)

1.13 Cliente de Weblate

Nuevo en la versión 2.7: There has been full wlc utility support ever since Weblate 2.7. If you are using an older version some incompatibilities with the API might occur.

1.13.1 Instalación

The Weblate Client is shipped separately and includes the Python module. To use the commands below, you need to install `wlc`:

```
pip3 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).

statistics

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.

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** Distinto en la versión 1.6: También se aceptan los archivos con la extensión **.ini**.

Archivo de configuración por proyecto

C:\Users\NOMBRE\AppData\weblate.ini Nuevo en la versión 1.6.

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/
```

(continué en la próxima página)

(proviene de la página anterior)

```
web: https://weblate.org/  
web_url: http://example.com/projects/hello/
```

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.

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):

- 2 GB de 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` 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 2.15-2: The setup has changed recently, priorly there was separate web server container, since 2.15-2 the web server is embedded in 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

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;
    listen [::]:443;

    server_name <SITE_URL>;
    ssl_certificate /etc/letsencrypt/live/<SITE>/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/<SITE>/privkey.pem;

    location / {
        proxy_set_header HOST $host;
        proxy_set_header X-Forwarded-Proto https;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Host $server_name;
        proxy_pass https://127.0.0.1:<EXPOSED_DOCKER_PORT>;
    }
}
```

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.

You can do this by sticking with the existing docker-compose and just pull the latest images and then restart:

```
docker-compose stop
docker-compose pull
docker-compose up
```

The Weblate database should be automatically migrated on first startup, and there should be no need for additional manual actions.

Nota: Upgrades across 3.0 are not supported by Weblate. If you are on 2.x series and want to upgrade to 3.x, first upgrade to the latest 3.0.1-x (at time of writing this it is the 3.0.1-7) image, 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. Please beware of PostgreSQL version changes in this case as it's not straightforward to upgrade the database, see [GitHub issue](#) for more info.

Admin sign in

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:
  UWSGI_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Ver también:

`CELERY_MAIN_OPTIONS`, `CELERY_NOTIFY_OPTIONS`, `CELERY_MEMORY_OPTIONS`, `CELERY_TRANSLATE_OPTIONS`, `CELERY_BACKUP_OPTIONS`, `CELERY_BEAT_OPTIONS`, `UWSGI_WORKERS`

Scaling horizontally

Nuevo en la versión 4.6.

Advertencia: This feature is a technology preview.

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 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.

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:

Admin sign in, 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:

Admin sign in, WEBLATE_ADMIN_PASSWORD, WEBLATE_ADMIN_NAME, WEBLATE_ADMIN_EMAIL

WEBLATE_SERVER_EMAIL

WEBLATE_DEFAULT_FROM_EMAIL

Configures the address for outgoing e-mails.

Ver también:

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_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_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

Configures GitHub username for GitHub pull-requests by changing *GITHUB_USERNAME*.

Ver también:

GitHub

WEBLATE_GITHUB_TOKEN

Nuevo en la versión 4.3.

Configures GitHub personal access token for GitHub pull-requests via API by changing *GITHUB_TOKEN*.

Ver también:

GitHub

WEBLATE_GITLAB_USERNAME

Configures GitLab username for GitLab merge-requests by changing *GITLAB_USERNAME*

Ver también:

GitLab

WEBLATE_GITLAB_TOKEN

Configures GitLab personal access token for GitLab merge-requests via API by changing *GITLAB_TOKEN*

Ver también:

GitLab

WEBLATE_PAGURE_USERNAME

Configures Pagure username for Pagure merge-requests by changing *PAGURE_USERNAME*

Ver también:

Pagure

WEBLATE_PAGURE_TOKEN

Configures Pagure personal access token for Pagure merge-requests via API by changing *PAGURE_TOKEN*

Ver también:

Pagure

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

Configures *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

Configures [LICENSE_FILTER](#).

WEBLATE_LICENSE_REQUIRED

Configures [LICENSE_REQUIRED](#)

WEBLATE_WEBSITE_REQUIRED

Configures [WEBSITE_REQUIRED](#)

WEBLATE_HIDE_VERSION

Configures [HIDE_VERSION](#).

WEBLATE_BASIC_LANGUAGES

Configures [BASIC_LANGUAGES](#).

WEBLATE_DEFAULT_AUTO_WATCH

Configures [DEFAULT_AUTO_WATCH](#).

WEBLATE_RATELIMIT_ATTEMPTS

WEBLATE_RATELIMIT_LOCKOUT

WEBLATE_RATELIMIT_WINDOW

Nuevo en la versión 4.6.

Configures rate limiter.

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

Configuración de traducción automática

WEBLATE_MT_APERTIUM_APY

Enables *Apertium* machine translation and sets *MT_APERTIUM_APY*

WEBLATE_MT_AWS_REGION

WEBLATE_MT_AWS_ACCESS_KEY_ID

WEBLATE_MT_AWS_SECRET_ACCESS_KEY

Configura la traducción automática de *AWS*.

```
environment:
  WEBLATE_MT_AWS_REGION: us-east-1
  WEBLATE_MT_AWS_ACCESS_KEY_ID: AKIAIOSFODNN7EXAMPLE
  WEBLATE_MT_AWS_SECRET_ACCESS_KEY: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

WEBLATE_MT_DEEPL_KEY

Activa la traducción automática de *DeepL* y establece *MT_DEEPL_KEY*

WEBLATE_MT_DEEPL_API_VERSION

Configura la versión de la API de *DeepL* que debe utilizarse; vea *MT_DEEPL_API_VERSION*.

WEBLATE_MT_GOOGLE_KEY

Activa *Google Translate* y establece *MT_GOOGLE_KEY*

WEBLATE_MT_MICROSOFT_COGNITIVE_KEY

Activa *Microsoft Cognitive Services Translator* y establece *MT_MICROSOFT_COGNITIVE_KEY*

WEBLATE_MT_MICROSOFT_ENDPOINT_URL

Establece *MT_MICROSOFT_ENDPOINT_URL*; observe que debe contener solo el nombre de dominio.

WEBLATE_MT_MICROSOFT_REGION

Establece *MT_MICROSOFT_REGION*

WEBLATE_MT_MICROSOFT_BASE_URL

Establece *MT_MICROSOFT_BASE_URL*

WEBLATE_MT_MODERNMT_KEY

Activa *ModernMT* y establece *MT_MODERNMT_KEY*.

WEBLATE_MT_MYMEMORY_ENABLED

Activa la traducción automática de *MyMemory* y establece *MT_MYMEMORY_EMAIL* a *WEBLATE_ADMIN_EMAIL*.

Ejemplo:

```
environment:
  WEBLATE_MT_MYMEMORY_ENABLED: 1
```

WEBLATE_MT_GLOSBE_ENABLED

Activa la traducción automática de *Glosbe*.

```
environment:
  WEBLATE_MT_GLOSBE_ENABLED: 1
```

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED

Activa la traducción automática de *Servicio terminológico de Microsoft*.

```
environment:
  WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED: 1
```

WEBLATE_MT_SAP_BASE_URL

WEBLATE_MT_SAP_SANDBOX_APIKEY

WEBLATE_MT_SAP_USERNAME

WEBLATE_MT_SAP_PASSWORD

WEBLATE_MT_SAP_USE_MT

Configura la traducción automática de *SAP Translation Hub*.

```
environment:
  WEBLATE_MT_SAP_BASE_URL: "https://example.hana.ondemand.com/translationhub/
  ↪api/v1/"
  WEBLATE_MT_SAP_USERNAME: "user"
  WEBLATE_MT_SAP_PASSWORD: "password"
  WEBLATE_MT_SAP_USE_MT: 1
```

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_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
```

(continué en la próxima página)

(proviene de la página anterior)

```

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**Activa la *Autenticación por GitHub*.**Bitbucket****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 *OAuth 2 de Google*.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

Activa la *OAuth 2 de GitLab*.

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

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_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Enables Slack authentication, see [Slack](#).

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

SAML Identity Provider settings, see [Autenticación por SAML](#).

Otras configuraciones de autenticación

WEBLATE_NO_EMAIL_AUTH

Desactiva la autenticación por correo electrónico al asignársele cualquier valor.

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_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](#).

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_TLS

Enables using SSL for Redis connection.

REDIS_VERIFY_SSL

Can be used to disable SSL certificate verification for Redis connection.

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

Mail server port, defaults to 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_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`.

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`.

Ver también:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_SSL`, `EMAIL_USE_TLS`

WEBLATE_EMAIL_BACKEND

Configures Django back-end to use for sending e-mails.

Ver también:

Configure e-mail sending, `EMAIL_BACKEND`

Site integration

WEBLATE_GET_HELP_URL

Configures `GET_HELP_URL`.

WEBLATE_STATUS_URL

Configures `STATUS_URL`.

WEBLATE_LEGAL_URL

Configures `LEGAL_URL`.

Error reporting

It is recommended to collect errors from the installation systematically, see [Collecting error reports](#).

To enable support for Rollbar, set the following:

ROLLBAR_KEY

Your Rollbar post server access token.

ROLLBAR_ENVIRONMENT

Your Rollbar environment, defaults to `production`.

To enable support for Sentry, set following:

SENTRY_DSN

Your Sentry DSN.

SENTRY_ENVIRONMENT

Your Sentry Environment (optional).

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, las comprobaciones, los complementos o las correcciones automáticas en funcionamiento

Nuevo en la versión 3.8-5.

Las variables que se enumeran a continuación permiten poner en funcionamiento o no las siguientes comprobaciones, complementos y correcciones automáticas:

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

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 matches the number of processors (except the backup worker, which is supposed to run only once).

Ejemplo:

```
environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```

Ver también:[Celery worker options](#), [Tareas en segundo plano con Celery](#)**UWSGI_WORKERS**

Configure how many uWSGI workers should be executed.

It defaults to number of processors + 1.

Ejemplo:

```
environment:
  UWSGI_WORKERS: 32
```

WEBLATE_SERVICE

Defines which services should be executed inside the container. Use this for [Scaling horizontally](#).

Following services are defined:

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 Translation memory Celery worker.

celery-notify Notifications Celery worker.

celery-translate Automatic translation Celery worker.

web Web server.

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/`. In the container it is mounted as `/app/data`.

The cache volume is mounted as `/app/cache` and is used to store static files. Its content is recreated on container startup and the volume can be mounted using ephemeral filesystem such as *tmpfs*.

Ver también:

[Docker volumes documentation](#)

Further configuration customization

You can further customize Weblate installation in the data volume, see *Volúmenes de contenedores Docker*.

Archivos de configuración personalizados

You can additionally override the configuration in `/app/data/settings-override.py` (see *Volúmenes de contenedores Docker*). This is executed at the end of built-in settings, after all environment settings are loaded, and you can adjust or override them.

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.

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_
      ↪ customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Adding own Python modules

Nuevo en la versión 3.8-5.

You can place own Python modules in `/app/data/python/` (see *Volúmenes de contenedores Docker*) and they can be then loaded by Weblate, most likely by using *Archivos de configuración personalizados*.

Ver también:

Personalizar Weblate

Select your machine - local or cloud providers

With Docker Machine you can create your Weblate deployment either on your local machine, or on any large number of cloud-based deployments on e.g. Amazon AWS, Greenhost, and many other providers.

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):

- 2 GB de 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 \
  libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-dev \
  libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev libacl1-dev libssl-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 tesseract-ocr libtesseract-dev liblibleptonica-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 nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
apt install apache2 libapache2-mod-wsgi

# Caching backend: Redis
apt install redis-server

# Database server: PostgreSQL
apt install postgresql postgresql-contrib

# SMTP server
apt install 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 --python=python3 ~/weblate-env
```

2. Active el entorno virtual para Weblate:

```
. ~/weblate-env/bin/activate
```

3. Instale Weblate y todas sus dependencias:

```
pip install Weblate
```

4. Instale el controlador de la base de datos:

```
pip install psycopg2-binary
```

5. Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [Dependencias opcionales](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseractocr
```

Configurar Weblate

Nota: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You can stick with shipped example for testing purposes, but you will want 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 production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [Running server](#) and [Serving static files](#)):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see [Compressing client assets](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Tareas en segundo plano con Celery](#) for more info:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

8. Start the development server (see [Running server](#) for 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/`.
- Login 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 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: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, 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):

- 2 GB de 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-
    <del>devel</del> \
    cairo-devel typelib-1_0-Pango-1_0 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
```

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 --python=python3 ~/weblate-env
```

2. Active el entorno virtual para Weblate:

```
. ~/weblate-env/bin/activate
```

3. Instale Weblate y todas sus dependencias:

```
pip install Weblate
```

4. Instale el controlador de la base de datos:

```
pip install psycopg2-binary
```

5. Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [Dependencias opcionales](#)):

```
pip install ruamel.yaml aedon boto3 zeep chardet tesseractocr
```

Configurar Weblate

Nota: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You can stick with shipped example for testing purposes, but you will want 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 production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [Running server](#) and [Serving static files](#)):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see [Compressing client assets](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Tareas en segundo plano con Celery](#) for more info:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

8. Start the development server (see [Running server](#) for 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/`.
- Login 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 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: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, 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):

- 2 GB de 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-
    <del>devel</del> \
    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](#)):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-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 --python=python3 ~/weblate-env
```

2. Active el entorno virtual para Weblate:

```
. ~/weblate-env/bin/activate
```

3. Instale Weblate y todas sus dependencias:

```
pip install Weblate
```

4. Instale el controlador de la base de datos:

```
pip install psycopg2-binary
```

5. Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [Dependencias opcionales](#)):

```
pip install ruamel.yaml aeidon boto3 zeep chardet tesseractocr
```

Configurar Weblate

Nota: Following steps assume virtualenv used by Weblate is active (what can be done by `./~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Adjust the values in the new `settings.py` file to your liking. You can stick with shipped example for testing purposes, but you will want 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 production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [Running server](#) and [Serving static files](#)):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see [Compressing client assets](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Tareas en segundo plano con Celery](#) for more info:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

8. Start the development server (see [Running server](#) for 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/`.
- Login 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 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: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, 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):

- 2 GB de 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
pip3 install virtualenv
```

Make sure pip will be able to find the libffi version provided by homebrew — this will be needed during the installation build step.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig"
```

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 --python=python3 ~/weblate-env
```

2. Active el entorno virtual para Weblate:

```
. ~/weblate-env/bin/activate
```

3. Instale Weblate y todas sus dependencias:

```
pip install Weblate
```

4. Instale el controlador de la base de datos:

```
pip install psycopg2-binary
```

5. Install wanted optional dependencies depending on features you intend to use (some might require additional system libraries, check [Dependencias opcionales](#)):

```
pip install ruamel.yaml aedon boto3 zeep chardet tesseract
```

Configurar Weblate

Nota: Following steps assume virtualenv used by Weblate is active (what can be done by `. ~/weblate-env/bin/activate`). In case this is not true, you will have to specify full path to **weblate** command as `~/weblate-env/bin/weblate`.

1. Copy the file `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` to `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.

2. Adjust the values in the new `settings.py` file to your liking. You can stick with shipped example for testing purposes, but you will want 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 production ready setup):

```
weblate migrate
```

4. Create the administrator user account and copy the password it outputs to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect static files for web server (see [Running server](#) and [Serving static files](#)):

```
weblate collectstatic
```

6. Compress JavaScript and CSS files (optional, see [Compressing client assets](#)):

```
weblate compress
```

7. Start Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. See [Tareas en segundo plano con Celery](#) for more info:

```
~/weblate-env/lib/python3.7/site-packages/weblate/examples/celery start
```

8. Start the development server (see [Running server](#) for 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/`.
- Login 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 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: Component name, VCS repository address and mask for finding translatable files. Weblate supports a wide range of formats including gettext PO files, Android resource strings, iOS string properties, Java properties or Qt Linguist files, 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. Siga las instrucciones de instalación correspondientes a su sistema primero:

- [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
```

4. Copy `weblate/settings_example.py` to `weblate/settings.py`.
5. Adjust the values in the new `settings.py` file to your liking. You can stick with shipped example for testing purposes, but you will want 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
weblate compilemessages
```

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.

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
```

Provisioning

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>
# ATTENTION! The following command is only optional and will permanently delete
→all of your data.
$ oc delete pvc -l app=<APPLICATION_NAME>

$ oc delete all -l app=weblate \
    && oc delete secret -l app=weblate \
    && oc delete configmap -l app=weblate \
    && oc delete pvc -l app=weblate
```

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.

Installing on 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
```

En función de la preparación y su experiencia, elija un método de instalación apropiado para usted:

- *Instalar con Docker*, recommended for production setups.
- Virtualenv installation, recommended for production setups:
 - *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*, recommended for development.
- *Instalar en OpenShift*
- *Installing on 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.celeryproject.org/>

Translate Toolkit <https://toolkit.translatehouse.org/>

translation-finder <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 admitir repositorios Mercurial) <https://www.mercurial-scm.org/>

phply (opcional para admitir PHP) <https://github.com/viraptor/phply>

tesseract (opcional para el OCR de capturas de pantalla) <https://github.com/sirfz/tesseract>

akismet (opcional para evitar *spam* en las sugerencias) <https://github.com/ubernostrum/akismet>

ruamel.yaml (opcional para *YAML files*) <https://pypi.org/project/ruamel.yaml/>

Zeep (opcional para *Servicio terminológico de Microsoft*) <https://docs.python-zeep.org/>

aeidon (opcional para *Archivos de subtítulos*) <https://pypi.org/project/aeidon/>

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 gir 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>

Build-time dependencies

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 (Braiiins) <michal.cihar@braiiins.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 (Braiiins) <michal.cihar@braiiins.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]
gpg:      aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:      aka "[jpeg image of size 8848]" [ultimate]
gpg:      aka "Michal Čihař (Braiiins) <michal.cihar@braiiins.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 Čihář <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 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.

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 -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 WITH SCHEMA weblate;
```

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

Consejo: 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 requiere MySQL o MariaDB, al menos en sus respectivas versiones 5.7.8 o 10.2.7.

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 emojis).
- 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.

Configuring Weblate to use 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.

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`

`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`

`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 migrate` to create the database structure. Now you should be able to create translation projects using the admin interface.

In case you want to run an installation non interactively, you can use `weblate migrate --noinput`, and then create an admin user using `createadmin` command.

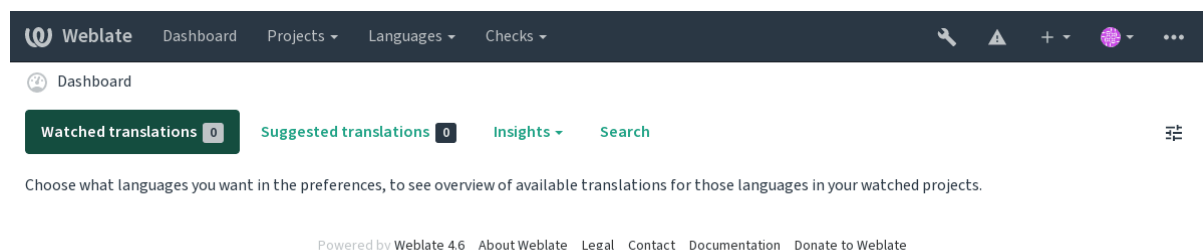
Once you are done, you should also check the *Performance report* in the admin interface, which will give you hints of potential non optimal configuration on your site.

Ver también:

Configuración, Lista de privilegios

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.

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",
        },
    },
}
```

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/examples/generate-secret-key` shipped with Weblate.

Ver también:

SECRET_KEY

Directorio del usuario

Distinto en la versión 2.1: This is no longer required, Weblate now stores all its data in `DATA_DIR`.

The home directory for the user running Weblate should exist and be writable by this user. This is especially needed if you want to use SSH to access private repositories, but Git might need to access this directory as well (depending on the Git version you use).

You can change the directory used by Weblate in `settings.py`, for example to set it to configuration directory under the Weblate tree:

```
os.environ["HOME"] = os.path.join(BASE_DIR, "configuration")
```

Nota: On Linux, and other UNIX like systems, the path to user's home directory is defined in `/etc/passwd`. Many distributions default to a non-writable directory for users used for serving web content (such as `apache`, `www-data` or `wwwrun`), so you either have to run Weblate under a different user, or change this setting.

Ver también:

Accessing repositories

Carga de plantillas

It is recommended to use a cached template loader for Django. It caches parsed templates and avoids the need to do parsing with every single request. You can configure it using the following snippet (the `loaders` setting is important here):

```
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [
            os.path.join(BASE_DIR, "templates"),
        ],
        "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",
            ],
            "loaders": [
```

(continué en la próxima página)

(proviene de la página anterior)

```

        (
            "django.template.loaders.cached.Loader",
            [
                "django.template.loaders.filesystem.Loader",
                "django.template.loaders.app_directories.Loader",
            ],
        ),
    ],
},
]

```

Ver también:`django.template.loaders.cached.Loader`**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 `commit_pending`.
- Updating component alerts (daily).
- Update remote branches (nightly), see `AUTO_UPDATE`.
- Translation memory backup to JSON (daily), see `dump_memory`.
- Fulltext and database maintenance tasks (daily and weekly tasks), see `cleanuptrans`.

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. For example when using Apache you might want to set it in `/etc/apache2/envvars`:

```

export LANG='en_US.UTF-8'
export LC_ALL='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 webserver 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

Distinto en la versión 2.4: Prior to version 2.4, Weblate didn't properly use the Django static files framework and the setup was more complex.

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*, *Deploying Django*, *Deploying 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/python3.7/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`):

```
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your_
↪setup.
server {
    listen 80;
    server_name weblate;
    # Not used
    root /var/www/html;

    location ~ ^/favicon.ico$ {
        # DATA_DIR/static/favicon.ico
        alias /home/weblate/data/static/favicon.ico;
        expires 30d;
    }

    location /static/ {
        # DATA_DIR/static/
        alias /home/weblate/data/static/;
        expires 30d;
    }

    location /media/ {
        # DATA_DIR/media/
        alias /home/weblate/data/media/;
        expires 30d;
    }

    location / {
        include uwsgi_params;
        # Needed for long running operations in admin interface
        uwsgi_read_timeout 3600;
        # Adjust based to uwsgi configuration:
        uwsgi_pass unix:///run/uwsgi/app/weblate/socket;
        # uwsgi_pass 127.0.0.1:8080;
    }
}
```

Configuration for uWSGI (also available as `weblate/examples/weblate.uwsgi.ini`):

```
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your_
↪setup.
[uwsgi]
plugins      = python3
master       = true
protocol     = uwsgi
socket       = 127.0.0.1:8080
wsgi-file    = /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/wsgi.
↪py
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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

# 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
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
#>env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your
#>setup.
#
```

(continué en la próxima página)

(proviene de la página anterior)

```

<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
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.7/site-packages/
    ↪weblate/wsgi.py process-group=weblate request-timeout=600
    WSGIPassAuthorization On

    <Directory /home/weblate/weblate-env/lib/python3.7/site-packages/weblate/>
        <Files wsgi.py>
            Require all granted
        </Files>
    </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`.

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
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your ↪
↪setup.
#
<VirtualHost *:443>
    ServerAdmin admin@weblate.example.org

```

(continué en la próxima página)

(proviene de la página anterior)

```

ServerName weblate.example.org

# DATA_DIR/static/favicon.ico
Alias /favicon.ico /home/weblate/data/static/favicon.ico

# DATA_DIR/static/
Alias /static/ /home/weblate/data/static/
<Directory /home/weblate/data/static/>
    Require all granted
</Directory>

# DATA_DIR/media/
Alias /media/ /home/weblate/data/media/
<Directory /home/weblate/data/media/>
    Require all granted
</Directory>

SSLEngine on
SSLCertificateFile /etc/apache2/ssl/https_cert.cert
SSLCertificateKeyFile /etc/apache2/ssl/https_key.pem
SSLProxyEngine On

ProxyPass /favicon.ico !
ProxyPass /static/ !
ProxyPass /media/ !

ProxyPass / http://localhost:8000/
ProxyPassReverse / http://localhost:8000/
ProxyPreserveHost On
</VirtualHost>

```

Ver también:[How to use Django with Gunicorn](#)**Running Weblate under path**

Nuevo en la versión 1.3.

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
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your_
↪setup.
#
<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/

```

(continué en la próxima página)

(proviene de la página anterior)

```

<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
WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.7/site-packages/
↪weblate/wsgi.py process-group=weblate request-timeout=600
WSGIPassAuthorization On

<Directory /home/weblate/weblate-env/lib/python3.7/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 process background tasks. 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](#)

For development, you might want to use eager configuration, which does process all tasks in place, but this will have performance impact on Weblate:

```

CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True

```

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*.

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"
```

(continué en la próxima página)

(proviene de la página anterior)

```
# Internal Weblate variable to indicate we're running inside Celery
CELERY_WORKER_RUNNING="1"
```

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 use `celery_queues` to see current length of Celery task queues. 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:

Configuration and defaults, Workers Guide, Daemonization, Monitoring and Management Guide, `celery_queues`

2.1.12 Monitoring Weblate

Weblate provides the `/healthz/` URL to be used in simple health checks, for example using Kubernetes.

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.

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 one is to dump the database on one server and import it on the new one. Alternatively you can use replication in case your database supports it.

The best approach is to use database native tools, as they are usually the most effective (e.g. **mysqldump** or **pg_dump**). If you want to migrate between different databases, the only option might be to use Django management to dump and import the database:

```
# Export current data
weblate dumpdata > /tmp/weblate.dump
# Import dump
weblate loaddata /tmp/weblate.dump
```

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*
- *Installing on 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>>. The setup will be adjusted during installation, see <<https://bitnami.com/stack/weblate/README.txt>> for more documentation.

Weblate Cloudron Package

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.

You may use your administration interface, or this button (it will bring you to your server):



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 upgrade steps integrated. There are no manual step besides pulling latest version.

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
```

With Git checkout you need to fetch new source code and update your installation:

```
cd weblate-src
git pull
# Update Weblate inside your virtualenv
. ~/weblate-env/bin/pip install -e .
# Install dependencies directly when not using virtualenv
pip install --upgrade -r requirements.txt
```

3. Upgrade configuration file, refer to `settings_example.py` or [Instrucciones específicas de las versiones](#) for needed steps.
4. Upgrade database structure:

```
weblate migrate --noinput
```

5. Collect updated static files (see [Running server](#) and [Serving static files](#)):

```
weblate collectstatic --noinput
```

6. Compress JavaScript and CSS files (optional, see [Compressing client assets](#)):

```
weblate compress
```

7. 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
```

8. Verify that your setup is sane (see also *Puesta en marcha de entorno de producción*):

```
weblate check --deploy
```

9. Restart celery worker (see *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 *DeepL* machine translation now defaults to v2 API, you might need to adjust `MT_DEEPL_API_VERSION` in case your current DeepL subscription does not support that.

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 *Ciente 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 addons 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: *Monolingual gettext* now uses both `msgid` and `msgctxt` when present. This will change identification of translation strings in such files breaking links to Weblate extended data such as screenshots or review states. Please make sure you commit pending changes in such files prior upgrading and it is recommended to force loading of affected component using `loadpo`. Increased minimal required version of `translate-toolkit` to address several file format issues.

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

Upgrade from 4.5 to 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`.
- API for creating components now automatically uses *URL internos de Weblate*, see `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

2.3.4 Upgrading from Python 2 to Python 3

Weblate no longer supports Python older than 3.5. 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 database than PostgreSQL, you should migrate to PostgreSQL as that will be the only supported database backend in the 4.0 release. 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 -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",
        # 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,
```

(continué en la próxima página)

(proviene de la página anterior)

```

    },
  },
  "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

Weblate fue creado originalmente como un sustituto de Pootle, de modo que es posible migrar cuentas de usuario desde esta plataforma. Puede volcar los usuarios en Pootle e importarlos mediante *importusers*.

2.4 Respaldo y trasladar Weblate

2.4.1 Copia de respaldo automatizada utilizando BorgBackup

Nuevo en la versión 3.9.

Weblate has built-in support for creating service backups using *BorgBackup*. Borg creates space-effective encrypted backups which can be safely stored in the cloud. The backups can be controlled in the management interface from the *Backups* tab.

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.

The backups using Borg are incremental and Weblate is configured to keep following backups:

- Copias de respaldo diarias para 14 días
- Copias de respaldo semanales para 8 semanas
- Copias de respaldo mensuales para 6 meses

Weblate
Dashboard
Projects
Languages
Checks

Manage / Backups

Backup process triggered

Weblate status
Backups
Translation memory
Performance report
SSH keys
Alerts
Repositories
Users
Appearance
Tools
Billing

Backup service: /tmp/tmpkrnjny6weblate

Backup service credentials

Backup repository

/tmp/tmpkrnjny6weblate

Passphrase

ZbTHOQCyWyE085zWc1xR1G(pr*UaPFg\$TDhVMYfbpBBCj)0A2g

The passphrase is used to encrypt the backups and is necessary to restore them.

SSH key

Download private key

The private key is needed to access the remote backup repository.

Deleted the oldest backups

Backup performed

Repository initialization

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 for remote SSH backups.

Add

Powered by Weblate 4.6
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Clave de cifrado de Borg

BorgBackup creates encrypted backups and you wouldn't be able to restore them without the passphrase. The passphrase is generated when adding a new backup service and you should copy it and keep it in a secure place.

If you are using *Weblate provisioned backup storage*, please backup your private SSH key too, as it's used to access your backups.

Ver también:

`borg init`

2.4.2 Weblate provisioned backup storage

The easiest way of backing up your Weblate instance is purchasing the [backup service at weblate.org](https://weblate.org/support/#backup). This is how you get it running:

1. Purchase the *Backup service* on <https://weblate.org/support/#backup>.
2. Enter the obtained key in the management interface, see *Integrating support*.
3. Weblate connects to the cloud service and obtains access info for the backups.
4. Turn on the new backup configuration from the *Backups* tab.
5. Backup your Borg credentials to be able to restore the backups, see *Clave de cifrado de Borg*.

Consejo: The manual step of turning everything on is there for your safety. Without your consent no data is sent to the backup repository obtained through the registration process.

2.4.3 Utilizar un almacenamiento personalizado para los respaldos

You can also use your own storage for the backups. SSH can be used to store backups in the remote destination, the target server needs to have **BorgBackup** installed.

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.

One option is to place backups into an existing volume, for example `/app/data/borgbackup`. This is an existing volume in the container.

You can also add a new container for the backups in the Docker Compose file for example by using `/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

In order to create the remote backups, you will have to install [BorgBackup](#) onto another server that's accessible via SSH. Make sure that it accepts the Weblate's client SSH key, i.e. the one it uses to connect to other servers.

Consejo: *Weblate provisioned backup storage* le ofrece copias de respaldo automatizadas.

Ver también:

Clave SSH de Weblate

2.4.4 Restaurar a partir de BorgBackup

1. Restore access to your backup repository and prepare your backup passphrase.
2. List all the backups on the server using `borg list REPOSITORY`.
3. Restore the desired backup to the current directory using `borg extract REPOSITORY::ARCHIVE`.
4. Restore the database from the SQL dump placed in the backup directory in the Weblate data dir (see *Datos volcados para las copias de respaldo*).
5. Copy the Weblate configuration (`backups/settings.py`, see *Datos volcados para las copias de respaldo*) to the correct location, see *Adjusting configuration*.
6. Copy the whole restored data dir to the location configured by `DATA_DIR`.

The Borg session might look like this:

```
$ 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`

2.4.5 Copia de respaldo manual

En función de lo que desee guardar, respalde los tipos de datos que Weblate almacena en cada sitio respectivo.

Consejo: If you are doing the manual backups, you might want to silence Weblate's warning about a lack of backups by adding `weblate.I028` to `SILENCED_SYSTEM_CHECKS` in `settings.py` or `WEBLATE_SILENCED_SYSTEM_CHECKS` for Docker.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

Base de datos

The actual storage location depends on your database setup.

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.

Native database backup

The recommended approach is to save a dump of the database using database-native tools such as `pg_dump` or `mysqldump`. It usually performs better than Django backup, and it restores complete tables with all their data.

You can restore this backup in a newer Weblate release, it will perform all the necessary migrations when running in `migrate`. Please consult *Actualizar Weblate* on more detailed info on how to upgrade between versions.

Django database backup

Alternatively, you can back up your database using Django's `dumpdata` command. That way the backup is database agnostic and can be used in case you want to change the database backend.

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 `migrate`.

Afterwards some entries will already be created in the database and you will have them in the database backup as well. The recommended approach is to delete such entries manually using the management shell (see *Invocar órdenes de gestión*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Archivos

If you have enough backup space, simply back up the whole `DATA_DIR`. This is a safe bet even if it includes some files you don't want. The following sections describe what you should back up and what you can skip in detail.

Datos volcados para las copias de respaldo

Almacenados en `DATA_DIR/backups`.

Weblate dumps various data here, and you can include these files for more complete backups. The files are updated daily (requires a running Celery beats server, see *Tareas en segundo plano con Celery*). Currently, this includes:

- Weblate settings as `settings.py` (there is also expanded version in `settings-expanded.py`).
- PostgreSQL database backup as `database.sql`.

The database backups are saved as plain text by default, but they can also be compressed or entirely skipped using `DATABASE_BACKUP`.

Repositorios de control de versiones

Stored in `DATA_DIR/vcs`.

The version control repositories contain a copy of your upstream repositories with Weblate changes. If you have *Push on commit* enabled for all your translation components, all Weblate changes are included upstream. No need to back up the repositories on the Weblate side as they can be cloned again from the upstream location(s) with no data loss.

Claves SSH y GPG

Stored in `DATA_DIR/ssh` and `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

Stored in `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

The Celery task queue might contain some info, but is usually not needed for a backup. At most you will lose updates not yet been processed to translation memory. It is recommended to perform the fulltext or repository update upon restoration anyhow, so there is no problem in losing these.

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
```

The string between the quotes after `XZ_OPT` allows you to choose your xz options, for instance the amount of memory used for compression; see <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
```

2.4.6 Restaurar una copia de respaldo manual

1. Restaure todos los datos de los que ha hecho copia de respaldo.
2. Actualice todos los repositorios mediante `updategit`.

```
weblate updategit --all
```

2.4.7 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

The default setup for Weblate is to use python-social-auth, a form on the website to handle registration of new users. After confirming their e-mail a new user can contribute or authenticate by using one of the third party services.

You can also turn off registration of new users using `REGISTRATION_OPEN`.

Los intentos de autenticación están sujetos a *Rate limiting*.

2.5.2 Dorsales de autenticación

The built-in solution of Django is used for authentication, including various social options to do so. Using it means you can import the user database of other Django-based projects (see *Migrar de Pootle*).

Django can additionally be set up to authenticate against other means too.

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.

Please check their documentation for generic configuration instructions in [Django Framework](#).

Nota: By default, Weblate relies on third-party authentication services to provide a validated e-mail address. If some of the services you want to use don't support this, please enforce e-mail validation on the Weblate side by configuring `FORCE_EMAIL_VALIDATION` for them. For example:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Ver también:

[Pipeline](#)

Enabling individual backends is quite easy, it's just a matter of adding an entry to the `AUTHENTICATION_BACKENDS` setting and possibly adding keys needed for a given authentication method. Please note that

some backends do not provide user e-mail by default, you have to request it explicitly, otherwise Weblate will not be able to properly credit contributions users make.

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

You need to register an OAuth application on GitHub and then tell Weblate all its secrets:

```
# 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://example.com/accounts/complete/github/`.

Nota: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see [Set correct site domain](#).

Ver también:

[GitHub](#)

Autenticación por Bitbucket

You need to register an application on Bitbucket and then tell Weblate all its secrets:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_VERIFIED_EMAILS_ONLY = True
```

Nota: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

Ver también:

Bitbucket

OAuth 2 de Google

Para utilizar OAuth 2 de Google, es necesario registrar una aplicación en <<https://console.developers.google.com/>> y activar la API de Google+.

The redirect URL is `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: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

Ver también:

Google

OAuth 2 de Facebook

As per usual with OAuth 2 services, you need to register your application with Facebook. Once this is done, you can set up Weblate to use it:

The redirect URL is `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: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

Ver también:

[Facebook](#)

OAuth 2 de GitLab

For using GitLab OAuth 2, you need to register an application on <https://gitlab.com/profile/applications>.

The redirect URL is `https://WEBLATE_SERVER/accounts/complete/gitlab/` and ensure you mark the `read_user` scope.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]

# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Nota: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

Ver también:

[GitLab](#)

Active Directory de Microsoft Azure

Weblate can be configured to use common or specific tenants for authentication.

The redirect URL is `https://WEBLATE_SERVER/accounts/complete/azuread-oauth2/` for common and `https://WEBLATE_SERVER/accounts/complete/azuread-tenant-oauth2/` for tenant-specific authentication.

```
# Azure AD common

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread.AzureADOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET = ""
```

```
# Azure AD Tenant

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread_tenant.AzureADTenantOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Nota: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see *Set correct site domain*.

Ver también:

Microsoft Azure Active Directory

Slack

For using Slack OAuth 2, you need to register an application on <https://api.slack.com/apps>.

The redirect URL is `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: Weblate provided callback URL during the authentication includes configured domain. In case you get errors about URL mismatch, you might want to fix this, see [Set correct site domain](#).

Ver también:

[Slack](#)

Desactivar la autenticación por contraseña

E-mail and password authentication can be turned off by removing `social_core.backends.email.EmailAuth` from `AUTHENTICATION_BACKENDS`. Always keep `weblate.accounts.auth.WeblateUserBackend` there, it is needed for core Weblate functionality.

Truco: You can still use password authentication for the admin interface, for users you manually create there. Just navigate to `/admin/`.

For example authentication using only the openSUSE Open ID provider can be achieved using the following:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Autenticación por contraseña

The default `settings.py` comes with a reasonable set of `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.

Additionally you can also install [django-zxcvbn-password](#) which gives quite realistic estimates of password difficulty and allows rejecting passwords below a certain threshold.

2.5.5 Autenticación por SAML

Nuevo en la versión 4.1.1.

Please follow the Python Social Auth instructions for configuration. Notable differences:

- Weblate supports single IDP which has to be called `weblate` in `SOCIAL_AUTH_SAML_ENABLED_IDPS`.
- The SAML XML metadata URL is `/accounts/metadata/saml/`.
- Following settings are automatically filled in: `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_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbn1+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
```

Ver también:

Configuring SAML in Docker, SAML

2.5.6 Autenticación LDAP

LDAP authentication can be best achieved using the *django-auth-ldap* package. You can install it via usual means:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

Advertencia: With *django-auth-ldap* older than 1.3.0 the *Asignaciones de grupo automático* will not work properly for newly created users.

Nota: There are some incompatibilities in the Python LDAP 3.1.0 module, which might prevent you from using that version. If you get error `AttributeError: "module" object has no attribute "_trace_level"`, downgrading python-ldap to 3.0.0 might help.

Once you have the package installed, you can hook it into the Django authentication:

```
# 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"
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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 `createadmin`).

Using bind password

If you can not use direct bind for authentication, you will need to use search, and provide a user to bind for the search. For example:

```
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
```

(continué en la próxima página)

(proviene de la página anterior)

```

    "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

```

(continúe en la próxima página)

(proviene de la página anterior)

```
# you can wrap the next two lines of code in a try/catch block.
user.email = attributes["email"]
user.save()
```

Ver también:

Django CAS NG

2.5.8 Configurar la autenticación de Django de terceros

Generally any Django authentication plugin should work with Weblate. Just follow the instructions for the plugin, just remember to keep the Weblate user backend installed.

Ver también:

Autenticación LDAP, Autenticación CAS

Typically the installation will consist of adding an authentication backend to `AUTHENTICATION_BACKENDS` and installing an authentication app (if there is any) into `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: Before Weblate 3.0, the privilege system was based on Django privilege system only, but is specifically built for Weblate now. If using anything older, please consult the documentation for the specific version you are using.

2.6.1 Simple access control

If you are not administrating the whole Weblate installation and just have access to manage certain projects (like on [Hosted Weblate](#)), your access control management options are limited to following settings. If you don't need any complex setup, those are sufficient for you.

Control de acceso al proyecto

Nota: Esta funcionalidad no está disponible para proyectos que usan el plan Libre en Hosted Weblate.

You can limit user's access to individual projects by selecting a different *Access control* setting. Available options are:

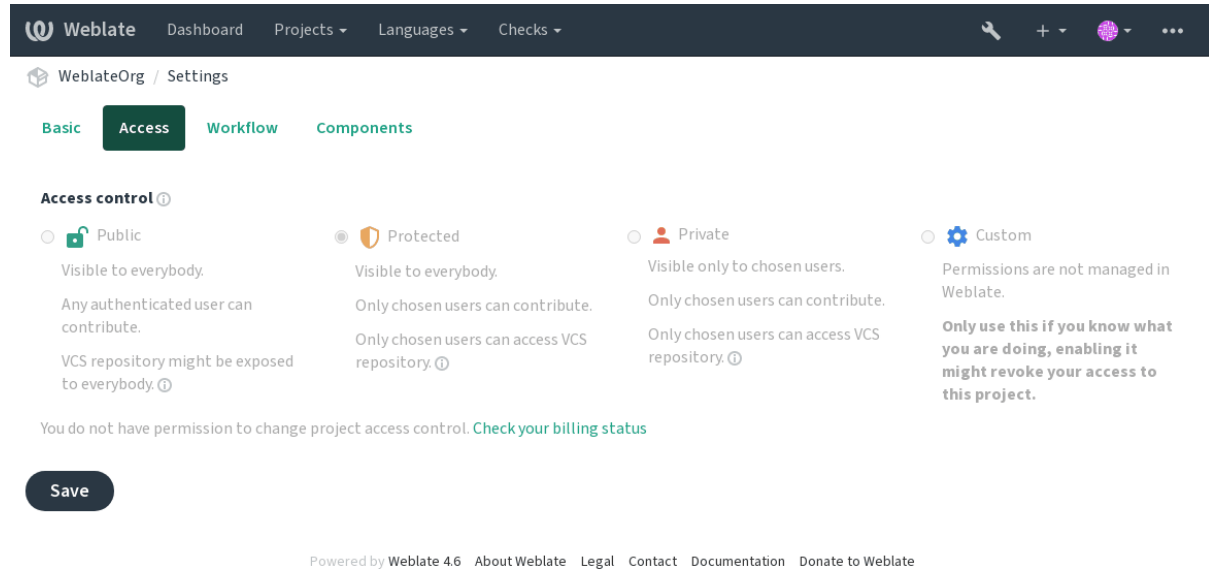
Público Publicly visible, translatable for all logged-in users.

Protegido Publicly visible, but translatable only for selected users.

Privado Visible and translatable only for selected users.

Personalizado *User management* features will be disabled; by default all users are forbidden to performed any actions on the project. You will have to set up all the permissions using *Control de acceso personalizado*.

Access control can be changed in the *Access* tab of the configuration (*Manage* ↓ *Settings*) of each respective project.



The default value can be changed by `DEFAULT_ACCESS_CONTROL`.

Nota: Even for *Private* projects, some info about your project will be exposed: statistics and language summary for the whole instance will include counts for all projects despite the access control setting. Your project name and other information can't be revealed through this.

Nota: The actual set of permissions available for users by default in *Public*, *Protected*, and *Private* projects can be redefined by Weblate instance administrator using *custom settings*.

Advertencia: By turning on *Custom* access control, Weblate will remove all *special groups* it has created for a selected project. If you are doing this without admin permission for the whole Weblate instance, you will instantly lose your access to manage the project.

Ver también:

Control de acceso

Gestionar el control de acceso por proyecto

Users with the *Manage project access* privilege (see *Lista de privilegios*) can manage users in projects with non-*Custom* access control. They can assign users to one of the following groups.

For *Public*, *Protected* and *Private* projects:

Administración Includes all permissions available for the project.

Review (only if *review workflow* is turned on) Puede aprobar traducciones durante la revisión.

For *Protected* and *Private* projects only:

Traducir Can translate the project and upload translations made offline.

Orígenes Can edit source strings (if allowed in the *project settings*) and source string info.

Idiomas Puede gestionar los idiomas traducibles (añadir o quitar traducciones).

Glosario Can manage glossary (add or remove entries, also upload).

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).

Sistema de control de versiones Puede gestionar el sistema de control de versiones y acceder al repositorio exportado.

Facturación Can access billing info and settings (see [Facturación](#)).

Unfortunately, it's not possible to change this predefined set of groups for now. Also this way it's not possible to give just some additional permissions to all users.

Nota: For non-*Custom* access control an instance of each group described above is actually defined for each project. The actual name of those groups will be `Project@Group`, also displayed in the Django admin interface this way. Although they can't be edited from Weblate user-interface.

The screenshot shows the Weblate web interface. The top navigation bar includes 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main header indicates 'WeblateOrg / Access control'. Below this, there are tabs for 'Users' (selected) and 'API access'. The 'Users' section displays a table with columns: Username, Full name, E-mail, Last login, Administration, Billing, Glossary, Languages, Memory, Screenshots, Sources, Translate, and VCS. A single user, 'testuser', is listed with a full name of 'Weblate Test', email 'weblate@example.org', and last login '25 seconds ago'. Below the table, a note states: 'Once all its permissions are removed, the user will be removed from the project.' At the bottom, there are two panels: 'Add a user' with a text input for 'User to add' and an 'Add' button, and 'Invite new user' with inputs for 'E-mail', 'Username', and 'Full name', along with an 'Invite' button. The footer contains links: 'Powered by Weblate 4.6', 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

These features are available on the *Access control* page, which can be accessed from the project's menu *Manage ↓ Users*.

New user invitation

Also, besides adding an existing user to the project, it is possible to invite new ones. Any new user will be created immediately, but the account will remain inactive until signing in with a link in the invitation sent via an e-mail. It is not required to have any site-wide privileges in order to do so, access management permission on the project's scope (e.g. a membership in the *Administration* group) would be sufficient.

Consejo: If the invited user missed the validity of the invitation, they can set their password using invited e-mail address in the password reset form as the account is created already.

Nuevo en la versión 3.11: It is possible to resend the e-mail for user invitations (invalidating any previously sent invitation).

The same kind of invitations are available site-wide from the *management interface* on the *Users* tab.

Per-project permission management

You can set your projects to *Protected* or *Private*, and *manage users* per-project in the Weblate user interface.

By default this prevents Weblate from granting access provided by *Users* and *Viewers default groups* due to these groups' own configuration. This doesn't prevent you from granting permissions to those projects site-wide by altering default groups, creating a new one, or creating additional custom settings for individual component as described in *Control de acceso personalizado* below.

One of the main benefits of managing permissions through the Weblate user interface is that you can delegate it to other users without giving them the superuser privilege. In order to do so, add them to the *Administration* group of the project.

2.6.2 Control de acceso personalizado

Nota: Esta funcionalidad no está disponible para proyectos que usan el plan Libre en Hosted Weblate.

The permission system is based on groups and roles, where roles define a set of permissions, and groups link them to users and translations, see *Users, roles, groups, and permissions* for more details.

The most powerful features of the Weblate's access control system for now are available only through the *Django admin interface*. You can use it to manage permissions of any project. You don't necessarily have to switch it to *Custom access control* to utilize it. However you must have superuser privileges in order to use it.

If you are not interested in details of implementation, and just want to create a simple-enough configuration based on the defaults, or don't have a site-wide access to the whole Weblate installation (like on *Hosted Weblate*), please refer to the *Simple access control* section.

Configuraciones comunes

This section contains an overview of some common configurations you may be interested in.

Site-wide permission management

To manage permissions for a whole instance at once, add users to appropriate *default groups*:

- *Users* (this is done by default by the *automatic group assignment*).
- *Reviewers* (if you are using *review workflow* with dedicated reviewers).
- *Managers* (if you want to delegate most of the management operations to somebody else).

You should keep all projects configured as *Public* (see *Control de acceso al proyecto*), otherwise the site-wide permissions provided by membership in the *Users* and *Reviewers* groups won't have any effect.

You may also grant some additional permissions of your choice to the default groups. For example, you may want to give a permission to manage screenshots to all the *Users*.

You can define some new custom groups as well. If you want to keep managing your permissions site-wide for these groups, choose an appropriate value for the *Project selection* (e.g. *All projects* or *All public projects*).

Permisos personalizados para idiomas, componentes o proyectos

You can create your own dedicated groups to manage permissions for distinct objects such as languages, components, and projects. Although these groups can only grant additional privileges, you can't revoke any permission granted by site-wide or per-project groups by adding another custom group.

Ejemplo:

If you want (for whatever reason) to allow translation to a specific language (lets say *Czech*) only to a closed set of reliable translators while keeping translations to other languages public, you will have to:

1. Remove the permission to translate *Czech* from all the users. In the default configuration this can be done by altering the *Users default group*.

Tabla 1: Group *Users*

Selección de idioma	<i>As defined</i>
Idiomas	All but <i>Czech</i>

2. Add a dedicated group for *Czech* translators.

Tabla 2: Group *Czech translators*

Roles	<i>Power users</i>
Selección del proyecto	<i>All public projects</i>
Selección de idioma	<i>As defined</i>
Idiomas	<i>Czech</i>

3. Add users you wish to give the permissions to into this group.

As you can see, permissions management this way is powerful, but can be quite a tedious job. You can't delegate it to another user, unless granting superuser permissions.

Users, roles, groups, and permissions

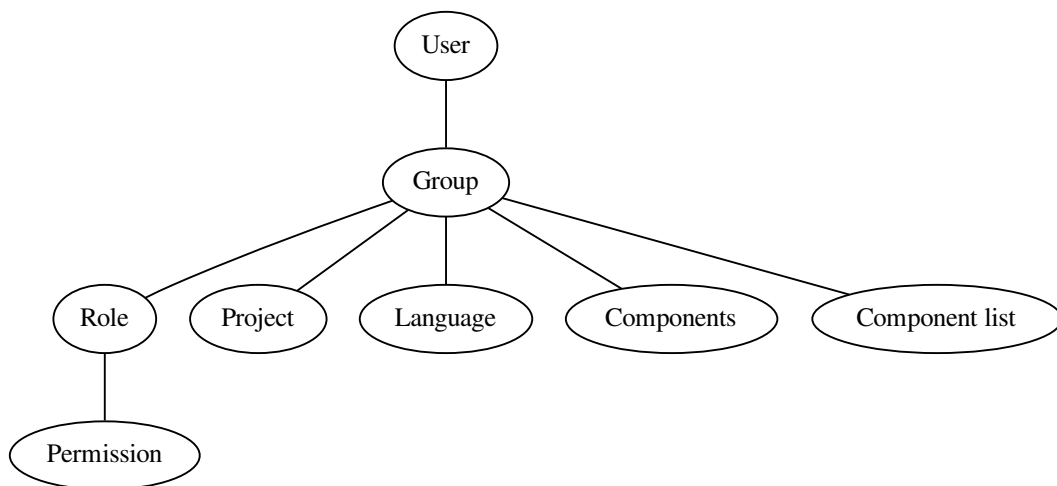
Los modelos de autenticación consisten de varios objetos:

Permiso Individual permission defined by Weblate. Permissions cannot be assigned to users. This can only be done through assignment of roles.

Role A role defines a set of permissions. This allows reuse of these sets in several places, making the administration easier.

Usuario User can belong to several groups.

Grupo Group connect roles, users, and authentication objects (projects, languages, and component lists).



Nota: A group can have no roles assigned to it, in that case access to browse the project by anyone is assumed (see below).

Access for browse to a project

A user has to be a member of a group linked to the project, or any component inside that project. Having membership is enough, no specific permissions are needed to browse the project (this is used in the default *Viewers* group, see *Lista de grupos*).

Access for browse to a component

A user can access unrestricted components once able to access the components' project (and will have all the permissions the user was granted for the project). With *Acceso restringido* turned on, access to the component requires explicit permissions for the component (or a component list the component is in).

Scope of groups

The scope of the permission assigned by the roles in the groups are applied by the following rules:

- If the group specifies any *Component list*, all the permissions given to members of that group are granted for all the components in the component lists attached to the group, and an access with no additional permissions is granted for all the projects these components are in. *Components* and *Projects* are ignored.
- If the group specifies any *Components*, all the permissions given to the members of that group are granted for all the components attached to the group, and an access with no additional permissions is granted for all the projects these components are in. *Projects* are ignored.
- Otherwise, if the group specifies any *Projects*, either by directly listing them or by having *Projects selection* set to a value like *All public projects*, all those permissions are applied to all the projects, which effectively grants the same permissions to access all projects *unrestricted components*.
- The restrictions imposed by a group's *Languages* are applied separately, when it's verified if a user has an access to perform certain actions. Namely, it's applied only to actions directly related to the translation process itself like reviewing, saving translations, adding suggestions, etc.

Consejo: Use *Language selection* or *Project selection* to automate inclusion of all languages or projects.

Ejemplo:

Let's say there is a project `foo` with the components: `foo/bar` and `foo/baz` and the following group:

Tabla 3: Group *Spanish Admin-Reviewers*

Roles	<i>Review Strings, Manage repository</i>
Componentes	<code>foo/bar</code>
Idiomas	<i>Spanish</i>

Members of that group will have following permissions (assuming the default role settings):

- General (browsing) access to the whole project `foo` including both components in it: `foo/bar` and `foo/baz`.
- Review strings in `foo/bar` Spanish translation (not elsewhere).
- Manage VCS for the whole `foo/bar` repository e.g. commit pending changes made by translators for all languages.

Asignaciones de grupo automático

On the bottom of the *Group* editing page in the *Django admin interface*, you can specify *Automatic group assignments*, which is a list of regular expressions used to automatically assign newly created users to a group based on their e-mail addresses. This assignment only happens upon account creation.

The most common use-case for the feature is to assign all new users to some default group. In order to do so, you will probably want to keep the default value (`^.*$`) in the regular expression field. Another use-case for this option might be to give some additional privileges to employees of your company by default. Assuming all of them use corporate e-mail addresses on your domain, this can be accomplished with an expression like `^.*@mycompany.com`.

Nota: Automatic group assignment to *Users* and *Viewers* is always recreated when upgrading from one Weblate version to another. If you want to turn it off, set the regular expression to `^$` (which won't match anything).

Nota: As for now, there is no way to bulk-add already existing users to some group via the user interface. For that, you may resort to using the *REST API*.

Default groups and roles

After installation, a default set of groups is created (see *Lista de grupos*).

These roles and groups are created upon installation. The built-in roles are always kept up to date by the database migration when upgrading. You can't actually change them, please define a new role if you want to define your own set of permissions.

Lista de privilegios

Facturación (vea *Facturación*) Ver información de facturación [*Administración, Facturación*]

Cambios Descargar cambios [*Administración*]

Comentarios Post comment [*Administration, Edit source, Power user, Review strings, Translate*]

Eliminar comentario [*Administración*]

Componente Editar configuración de componentes [*Administración*]

Lock component, preventing translations [*Administration*]

Glosario Añadir entrada de glosario [*Administración, Gestionar glosario, Usuario avanzado*]

Editar entrada de glosario [*Administración, Gestionar glosario, Usuario avanzado*]

Eliminar entrada de glosario [*Administración, Gestionar glosario, Usuario avanzado*]

Cargar entradas de glosario [*Administración, Gestionar glosario, Usuario avanzado*]

Sugerencias automáticas Use automatic suggestions [*Administration, Edit source, Power user, Review strings, Translate*]

Memoria de traducción Edit translation memory [*Administration, Manage translation memory*]

Delete translation memory [*Administration, Manage translation memory*]

Proyectos Editar configuración del proyecto [*Administración*]

Gestionar acceso al proyecto [*Administración*]

Informes Download reports [*Administration*]

Capturas de pantalla Add screenshot [*Administration, Manage screenshots*]

Edit screenshot [*Administration, Manage screenshots*]

Delete screenshot [*Administration, Manage screenshots*]

Cadenas de origen Edit additional string info [*Administration, Edit source*]

Cadenas Add new string [*Administration*]

Remove a string [*Administration*]

Ignore failing check [*Administration, Edit source, Power user, Review strings, Translate*]

Edit strings [*Administration, Edit source, Power user, Review strings, Translate*]

Review strings [*Administration, Review strings*]

Edit string when suggestions are enforced [*Administration, Review strings*]

Edit source strings [*Administration, Edit source, Power user*]

Sugerencias Accept suggestion [*Administration, Edit source, Power user, Review strings, Translate*]

Add suggestion [*Administration, Edit source, Add suggestion, Power user, Review strings, Translate*]

Delete suggestion [*Administration, Power user*]

Vote on suggestion [*Administration, Edit source, Power user, Review strings, Translate*]

Traducciones Add language for translation [*Administration, Power user, Manage languages*]

Perform automatic translation [*Administration, Manage languages*]

Delete existing translation [*Administration, Manage languages*]

Add several languages for translation [*Administration, Manage languages*]

Cargas Define author of uploaded translation [*Administration*]

Overwrite existing strings with upload [*Administration, Edit source, Power user, Review strings, Translate*]

Upload translations [*Administration, Edit source, Power user, Review strings, Translate*]

Sistema de control de versiones Access the internal repository [*Administration, Access repository, Power user, Manage repository*]

Commit changes to the internal repository [*Administration, Manage repository*]

Push change from the internal repository [*Administration, Manage repository*]

Reset changes in the internal repository [*Administration, Manage repository*]

View upstream repository location [*Administration, Access repository, Power user, Manage repository*]

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

Gestionar grupos

Administrar usuarios

Gestionar roles

Gestionar anuncios

Gestionar memoria de traducción

Gestionar listas de componentes

Nota: Site-wide privileges are not granted to any default role. These are powerful and quite close to superuser status. Most of them affect all projects in your Weblate installation.

Lista de grupos

The following groups are created upon installation (or after executing `setupgroups`) and you are free to modify them. The migration will, however, re-create them if you delete or rename them.

Guests Define los permisos que tendrán los usuarios no autenticados.

Este grupo contiene solo usuarios anónimos (vea `ANONYMOUS_USER_NAME`).

You can remove roles from this group to limit permissions for non-authenticated users.

Default roles: *Add suggestion, Access repository*

Lectores This role ensures visibility of public projects for all users. By default, all users are members of this group.

By default, *automatic group assignment* makes all new accounts members of this group when they join.

Default roles: none

Users Grupo predeterminado para todos los usuarios.

By default, *automatic group assignment* makes all new accounts members of this group when they join.

Default roles: *Power user*

Revisores Group for reviewers (see *Flujos de trabajo de traducción*).

Default roles: *Review strings*

Supervisores Grupo para administradores.

Default roles: *Administration*

Advertencia: Never remove the predefined Weblate groups 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 Additional access restrictions

If you want to use your Weblate installation in a less public manner, i.e. allow new users on an invitational basis only, it can be done by configuring Weblate in such a way that only known users have an access to it. In order to do so, you can set `REGISTRATION_OPEN` to `False` to prevent registrations of any new users, and set `REQUIRE_LOGIN` to `/.*` to require logging-in to access all the site pages. This is basically the way to lock your Weblate installation.

Consejo: You can use built-in *invitations* to add new users.

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 *File mask* 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.

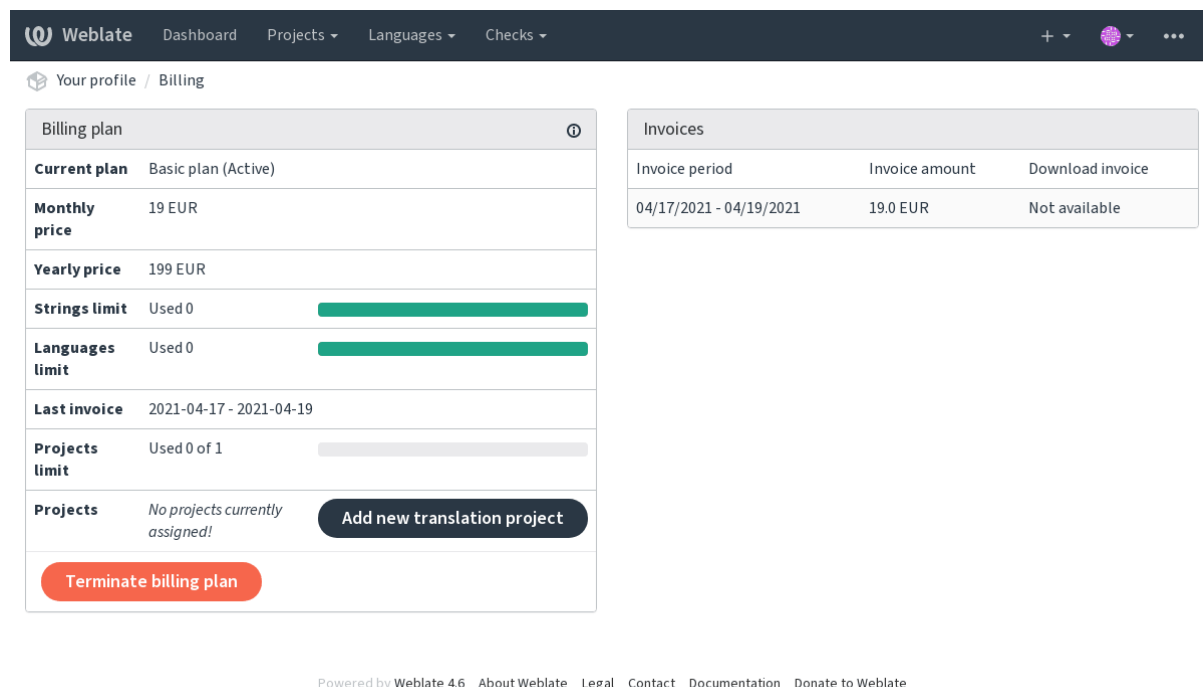
2.7.2 Añadir proyectos y componentes de traducción

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 screenshot shows the Weblate interface with a dark navigation bar at the top containing 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below the navigation bar, the breadcrumb 'Your profile / Billing' is visible. The main content area is divided into two panels. The left panel, titled 'Billing plan', contains a table with the following data:

Billing plan	
Current plan	Basic plan (Active)
Monthly price	19 EUR
Yearly price	199 EUR
Strings limit	Used 0
Languages limit	Used 0
Last invoice	2021-04-17 - 2021-04-19
Projects limit	Used 0 of 1
Projects	No projects currently assigned! <button>Add new translation project</button>
<button>Terminate billing plan</button>	

The right panel, titled 'Invoices', contains a table with the following data:

Invoice period	Invoice amount	Download invoice
04/17/2021 - 04/19/2021	19.0 EUR	Not available

At the bottom of the page, a footer line reads: 'Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate'.

The project creation can be initiated from there, or using the menu in the navigation bar, filling in basic info about the translation project to complete addition of it:

Create project

Add new translation project

Project name ⓘ

WeblateOrg

Display name

URL slug ⓘ

weblateorg

Name used in URLs and filenames.

Project website ⓘ

https://weblate.org/

Main website of translated project.

Translation instructions ⓘ

https://weblate.org/contribute/

You can use Markdown and mention users by @username.

Billing ⓘ

Weblate Test (Basic plan)

Save

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

After creating the project, you are taken directly to the project page:

Weblate

Dashboard Projects Languages Checks

WeblateOrg

translated 100%

Components Languages Info Search Insights Files Tools Manage Share

Watching

Nothing to list here.

Add new translation component

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Creating a new translation component can be initiated via a single click there. The process of creating a component is multi-staged and automatically detects most translation parameters. There are several approaches to creating component:

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.

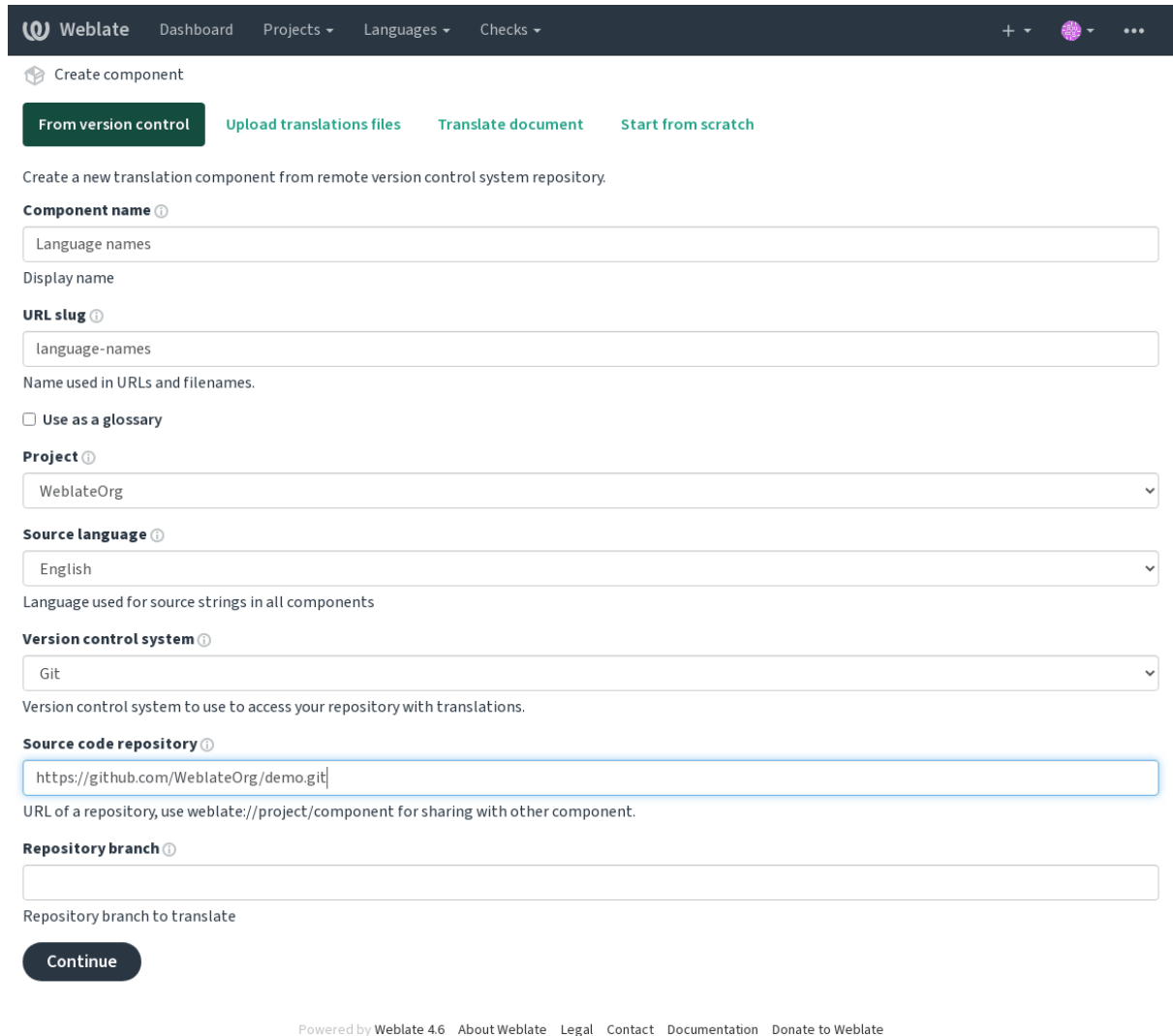
Cargar archivos de traducción Upload translation files to Weblate in case you do not have version control or do not want to integrate it with Weblate. You can later update the content using the web interface or [API REST de Weblate](#).

Traducir documento Cargue un documento único y tradúzcalo.

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:

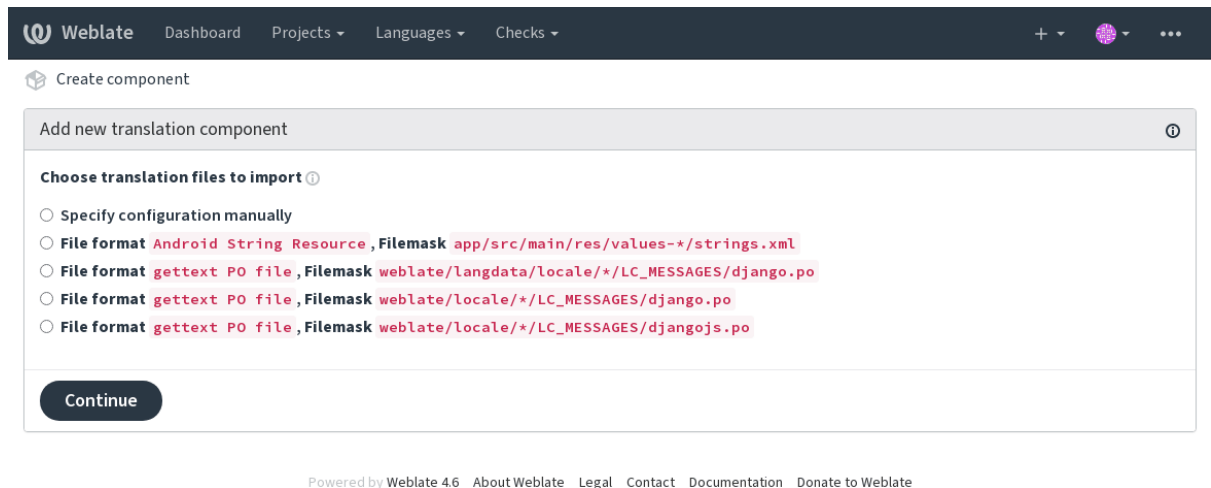


The screenshot shows the 'Create component' form in the Weblate web interface. The form is titled 'Create component' and has a dark header bar with the Weblate logo and navigation links: Dashboard, Projects, Languages, and Checks. Below the header, there are four tabs: 'From version control' (selected), 'Upload translations files', 'Translate document', and 'Start from scratch'. The main form area contains several fields and sections:

- Component name**: A text input field with the placeholder 'Language names'.
- Display name**: A text input field.
- URL slug**: A text input field with the value 'language-names'.
- Use as a glossary**: A checkbox that is currently unchecked.
- Project**: A dropdown menu with 'WeblateOrg' selected.
- Source language**: A dropdown menu with 'English' selected.
- Version control system**: A dropdown menu with 'Git' selected.
- Source code repository**: A text input field with the value 'https://github.com/WeblateOrg/demo.git'.
- Repository branch**: A text input field.

At the bottom of the form is a 'Continue' button. Below the form, there is a footer bar with the text 'Powered by Weblate 4.6' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

En la próxima página verá una lista de recursos traducibles detectados:



The screenshot shows the 'Add new translation component' form in the Weblate web interface. The form is titled 'Add new translation component' and has a dark header bar with the Weblate logo and navigation links: Dashboard, Projects, Languages, and Checks. Below the header, there are four tabs: 'From version control' (selected), 'Upload translations files', 'Translate document', and 'Start from scratch'. The main form area contains several fields and sections:

- Choose translation files to import**: A section with a title and a list of radio buttons.
- Specify configuration manually**: A radio button that is currently selected.
- File format**: A list of file formats with their corresponding file masks and paths:

- Android String Resource**: Filemask `app/src/main/res/values-*/strings.xml`
- gettext PO file**: Filemask `weblate/langdata/locale/*/LC_MESSAGES/django.po`
- gettext PO file**: Filemask `weblate/locale/*/LC_MESSAGES/django.po`
- gettext PO file**: Filemask `weblate/locale/*/LC_MESSAGES/djangojs.po`

At the bottom of the form is a 'Continue' button. Below the form, there is a footer bar with the text 'Powered by Weblate 4.6' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Como último paso, revise la información del componente de traducción y supla datos opcionales:

Powered by [Weblate 4.6](#) [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.

«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*.

Instrucciones de traducción

URL to more site with more detailed instructions for translators.

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.

Default value is determined by *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.

Default value is determined by `DEFAULT_SHARED_TM`.

Control de acceso

Configure per project access control, see *Control de acceso al proyecto* for more details.

Default value can be changed by `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*, *Language definitions*

Alias de idiomas

Define language codes mapping when importing translations into Weblate. Use this when language codes are inconsistent in your repositories and you want to get a consistent view in Weblate or in case you want to use non-standard naming of your translation files.

The typical use case might be mapping American English to English: `en_US:en`

Multiple mappings to be separated by comma: `en_GB:en, en_US:en`

Using non standard code: `ia_FOO:ia`

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:

Parsing language codes

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: Es aconsejable mantener un tamaño razonable para los componentes de traducción. Divídala en secciones que tengan sentido para su proyecto (por ejemplo, por aplicaciones o complementos, capítulos 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. This setting is used only for *Git* and *Mercurial* and push support is turned off for these when this is empty.

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, 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.

Rama a la que enviar

Branch for pushing changes, leave empty to use *Rama del repositorio*.

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

File mask

Mask of files to translate, including path. It should include one «*» replacing language code (see *Language definitions* 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)»?

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, e.g. `.pot` file with `gettext`.

Consejo: In many monolingual formats Weblate starts with blank 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, *Añadir traducciones nuevas*, *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`.

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, [Añadir 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.

Ver también:

[Añadir traducciones nuevas](#), [Código de idioma](#), [Parsing language codes](#)

Estilo de fusión

You can configure how updates from the upstream repository are handled. This might not be supported for some VCSs. See *Fusionar o cambiar base* for more details.

Default value can be changed by `DEFAULT_MERGE_STYLE`.

Mensajes de consigna, de adición, de eliminación, de fusión y de complemento

Message used when committing a translation, see *Marcación de plantilla*.

Default value can be changed by `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE`, `DEFAULT_MERGE_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 `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

Regular expression used to filter the translation when scanning for filemask. It can be used to limit the list of languages managed by 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	<code>^(cs de es)\$</code>
Excluir idiomas	<code>^(?! (it fr)\$) .+\$</code>
Filter two letter codes only	<code>^[.]+\$</code>
Excluir archivos no lingüísticos	<code>^(?! (blank)\$) .+\$</code>
Incluir todos los archivos (predeterminado)	<code>^[^.] +\$</code>

Expresión regular de variantes

Regular expression used to determine the variants of a string, see variants.

Nota: Los propietarios y los gestores del proyecto pueden editar la mayoría de los demás campos, 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.

Acceso restringido

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

You can choose additional projects where the component will be visible. Useful for shared libraries which you use in several projects.

Nota: Sharing a component doesn't change its access control. It only makes it visible when browsing other projects. Users still need access to the actual component to browse or translate it.

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*
- **Algunos complementos**
 - *Detección de componentes*
 - *Generador de estadísticas*
 - *Ejecutar secuencias de órdenes de un 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 addon, available only in the addon 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 {{file-
name|parentdir}}

```

You can combine them with filters:

```
{{ component|title }}
```

You can use conditions:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

There is additional tag available for replacing characters:

```
{% replace component "-" " " %}
```

You can combine it with filters:

```
{% replace component|capfirst "-" " " %}
```

There are also additional filter to manipulate with filenames:

```

Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|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 *import_project* or *import_json*, or by installing the *Detección de componentes* addon.

To use the addon, 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 addon on this component.

For the management commands, you need to create a project which will contain all components and then run *import_project* or *import_json*.

Ver también:

Órdenes de gestión, Detección de componentes

2.8 Language definitions

To present different translations properly, info about language name, text direction, plural definitions and language code is needed.

2.8.1 Parsing language codes

While parsing translations, Weblate attempts to map language code (usually the ISO 639-1 one) 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:

- Case insensitive lookups.
- Normalizing underscores and dashes.
- Looking up built-in language aliases.
- 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 *Changing language definitions*) 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, *Añadir traducciones nuevas*

2.8.2 Changing language definitions

You can change language definitions in the languages interface (`/languages/` URL).

While editing, make sure all fields are correct (especially plurals and text direction), otherwise translators will be unable to properly edit those translations.

2.8.3 Built-in language definitions

Definitions for about 600 languages are included in Weblate and the list is extended in every release. Whenever Weblate is upgraded (more specifically whenever **weblate migrate** is executed, see *Instrucciones de actualización genéricas*) the database of languages is updated to include all language definitions shipped in Weblate.

This feature can be disabled using `UPDATE_LANGUAGES`. You can also enforce updating the database to match Weblate built-in data using `setuplang`.

Ver también:

The language definitions are in the [weblate-language-data repository](#).

2.8.4 Ambiguous language codes and macrolanguages

In many cases it is not a good idea to use macro language 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 macro languages.

Ver también:

[Macrolanguages definition](#), [List of macrolanguages](#)

2.8.5 Language definitions

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:

[Parsing language codes](#), [Añadir traducciones nuevas](#)

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](#)

2.8.6 Añadir traducciones nuevas

Distinto en la versión 2.18: In versions prior to 2.18 the behaviour of adding new translations was file format specific.

Weblate puede iniciar automáticamente traducciones nuevas en todos los formatos de archivo.

Some formats expect to start with an empty file and only translated strings to be included (for example [Recursos de cadenas de Android](#)), while others expect to have all keys present (for example [gettext de GNU](#)). In some situations this really doesn't depend on the format, but rather on the framework you use to handle the translation (for example with [Archivos JSON](#)).

When you specify *Plantilla para traducciones nuevas* in *Configuración de componentes*, Weblate will use this file to start new translations. Any exiting translations will be removed from the file when doing so.

When *Plantilla para traducciones nuevas* is empty and the file format supports it, an empty file is created where new strings will be added once they are translated.

La opción *Estilo de código de idioma* le permite personalizar los códigos de idioma que se utilizarán en los nombres de archivo generados:

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`).

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 Android Only used in Android apps, produces language codes like `pt-rBR`.

Estilo Java Used by Java—mostly BCP with legacy codes for Chinese.

Additionally, any mappings defined in *Alias de idiomas* are applied in reverse.

Nota: Weblate recognizes any of these when parsing translation files, the above settings only influences how new files are created.

Ver también:

Código de idioma, Parsing language codes

2.9 Regionalización continua

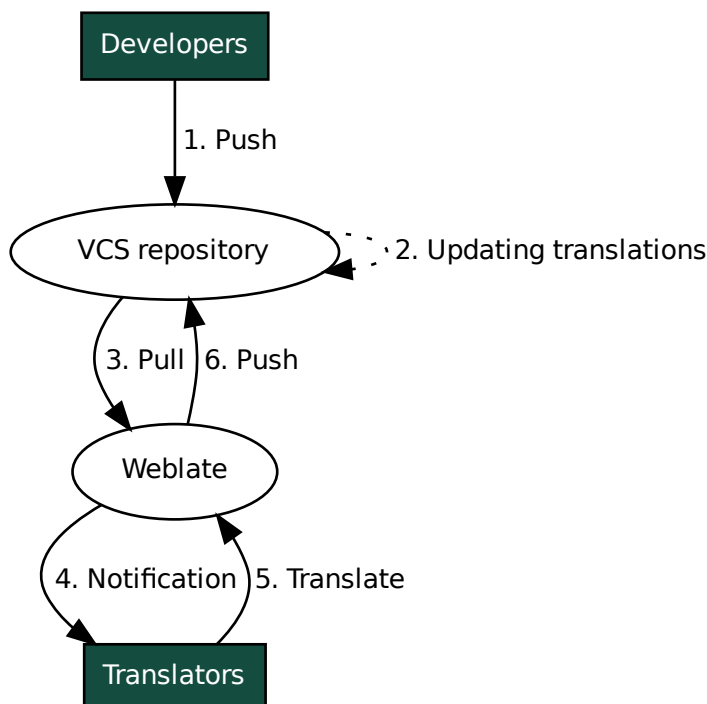
There is infrastructure in place so that your translation closely follows development. This way translators can work on translations the entire time, instead of working through huge amount of new text just prior to release.

Ver también:

`/dev/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 (this depends on the file format, see *¿Por qué Weblate sigue mostrando cadenas de traducción viejas luego de que he actualizado la plantilla?*).
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*
- 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 `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:

The screenshot shows the GitHub 'Add webhook' page for the repository 'WeblateOrg / hello'. The left sidebar contains navigation links: Options, Collaborators & teams, Branches, Webhooks (highlighted), Integrations & services, Deploy keys, and Alerts. The main content area is titled 'Webhooks / Add webhook' and includes the following fields and options:

- Payload URL:** A text input field containing 'https://hosted.weblate.org/hooks/github/'.
- Content type:** A dropdown menu set to 'application/x-www-form-urlencoded'.
- Secret:** An empty text input field.
- SSL verification:** A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' with a red button to 'Disable SSL verification'.
- Which events would you like to trigger this webhook?:** Three radio button options: 'Just the push event.' (selected), 'Send me everything.', and 'Let me select individual events.'
- Active:** A checked checkbox with the text 'We will deliver event details when this hook is triggered.'
- Add webhook:** A green button at the bottom.

The footer of the page shows copyright information for GitHub, Inc. and links to Terms, Privacy, Security, Status, Help, Contact GitHub, API, Training, Shop, Blog, and About.

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/`.

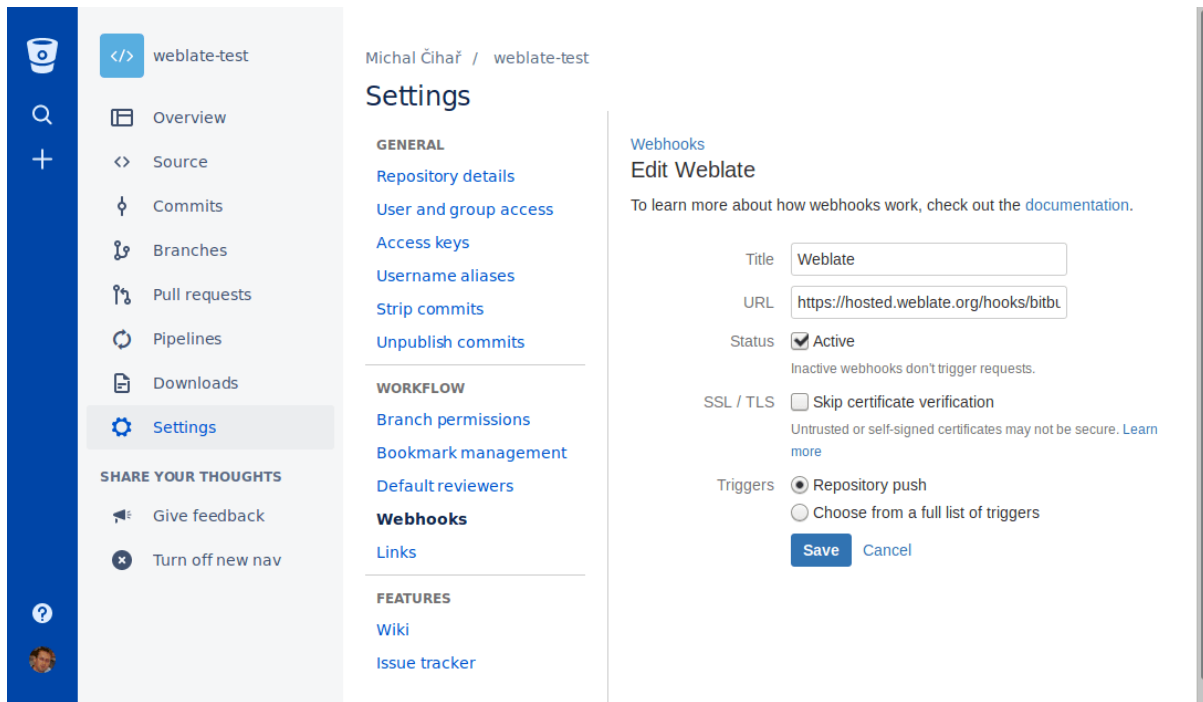
You can leave other values at default settings (Weblate can handle both content types and consumes just the *push* event).

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/`).

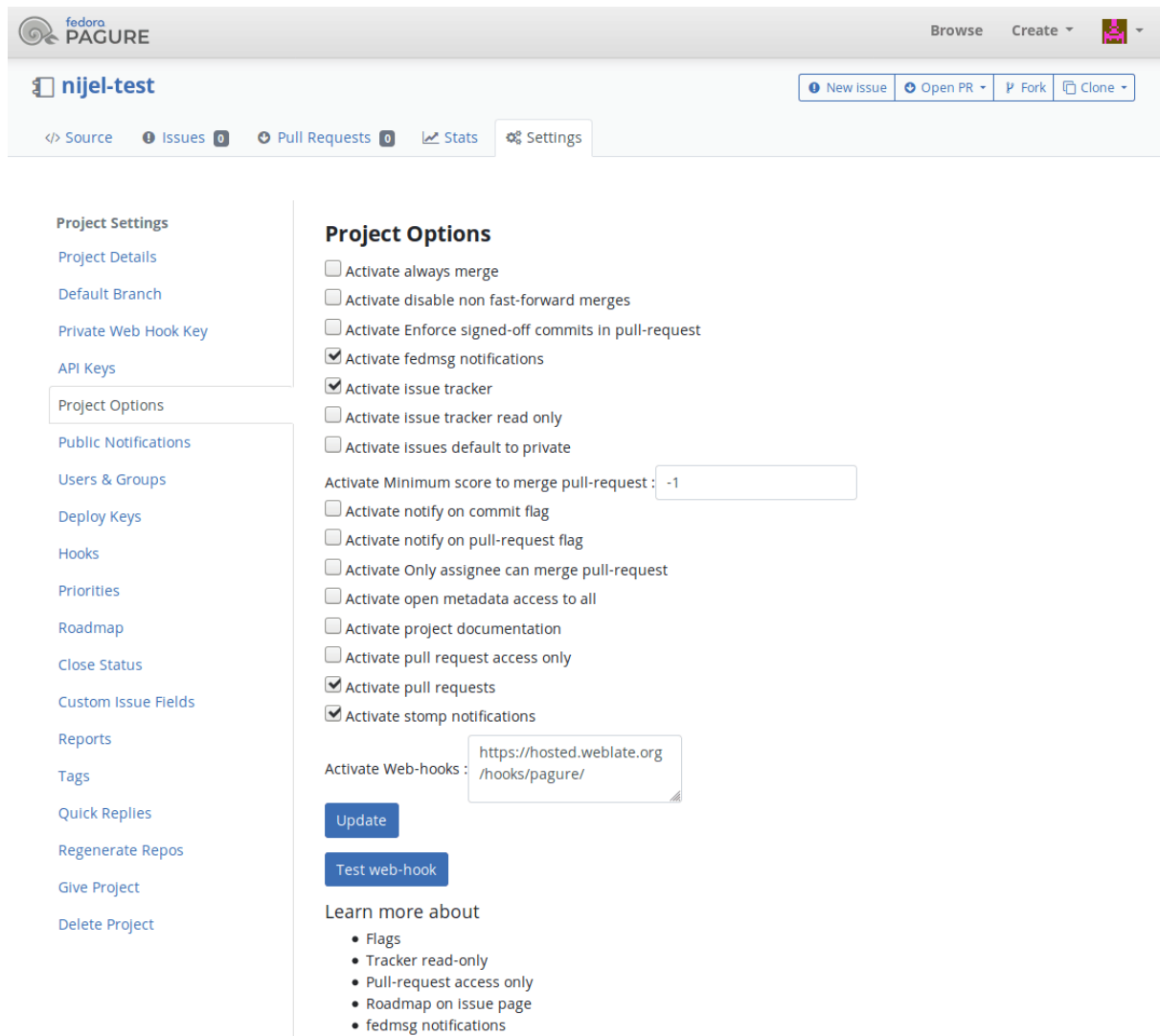
Ver también:

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*:



The screenshot shows the Weblate interface for a project named 'nijel-test'. The top navigation bar includes 'Browse', 'Create', and a user profile icon. Below the project name, there are links for 'New Issue', 'Open PR', 'Fork', and 'Clone'. The left sidebar lists various settings categories, with 'Project Options' currently selected. The main content area displays the 'Project Options' settings, which include a list of checkboxes for enabling various features like 'Activate always merge', 'Activate disable non fast-forward merges', 'Activate Enforce signed-off commits in pull-request', 'Activate fedmsg notifications', 'Activate Issue tracker', 'Activate Issue tracker read only', 'Activate Issues default to private', 'Activate Minimum score to merge pull-request', 'Activate notify on commit flag', 'Activate notify on pull-request flag', 'Activate Only assignee can merge pull-request', 'Activate open metadata access to all', 'Activate project documentation', 'Activate pull request access only', 'Activate pull requests', and 'Activate stomp notifications'. There are also buttons for 'Update' and 'Test web-hook'. At the bottom, there is a section 'Learn more about' with links to 'Flags', 'Tracker read-only', 'Pull-request access only', 'Roadmap on Issue page', and 'fedmsg notifications'.

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 *GitHub*, *GitLab*, *Pagure* pull requests or *Gerrit* reviews, you can activate these by choosing *GitHub*, *GitLab*, *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	<i>Sistema de control de versiones</i>	<i>URL de envío al repositorio</i>	<i>Rama a la que enviar</i>
No push	<i>Git</i>	<i>empty</i>	<i>empty</i>
Enviar directamente	<i>Git</i>	URL SSH	<i>empty</i>
Enviar en una rama separada	<i>Git</i>	URL SSH	Nombre de la rama
GitHub pull request from fork	<i>GitHub</i>	<i>empty</i>	<i>empty</i>
GitHub pull request from branch	<i>GitHub</i>	SSH URL ¹	Nombre de la rama
GitLab merge request from fork	<i>GitLab</i>	<i>empty</i>	<i>empty</i>
GitLab merge request from branch	<i>GitLab</i>	SSH URL ^{Página 60, 1}	Nombre de la rama
Pagure merge request from fork	<i>Pagure</i>	<i>empty</i>	<i>empty</i>
Pagure merge request from branch	<i>Pagure</i>	SSH URL ^{Página 60, 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.

☒ **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

Required approving reviews: 1 ▾

☐ **Dismiss stale pull request approvals when new commits are pushed**

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

☐ **Require review from Code Owners**

Require an approved review in pull requests including files with a designated code owner.

☒ **Restrict who can dismiss pull request reviews**

Specify people or teams allowed to dismiss pull request reviews.

Search for people or teams

People and teams that can dismiss reviews.

 **Organization and repository administrators**
These members can always dismiss.

 **weblate**
Weblate push user

×

2.9.3 Fusionar o cambiar base

By default, Weblate merges the upstream repository into its own. This is the safest way in case you also access the underlying repository by other means. In case you don't need this, you can enable rebasing of changes on upstream, which will produce a history with fewer merge commits.

Nota: Rebasing can cause you trouble in case of complicated merges, so carefully consider whether or not you want to enable them.

2.9.4 Interactuar con otros

Weblate facilita la interacción con otras herramientas mediante su API.

Ver también:

API REST de Weblate

2.9.5 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](#)).

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.
- 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* addon 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:

```
CELERY_BEAT_SCHEDULE = {
    # Unconditionally commit all changes every 2 minutes
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Ommiting hours will honor per component settings,
        # otherwise components with no changes older than this
        # won't be committed
        "kwargs": {"hours": 0},
        # How frequently to execute the job in seconds
        "schedule": 120,
    }
}
```

2.9.6 Procesar el repositorio con secuencias

The way to customize how Weblate interacts with the repository is *Complementos*. Consult *Ejecutar secuencias de órdenes de un complemento* for info on how to execute external scripts through addons.

2.9.7 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 translation propagation 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 addon (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 screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb 'WeblateOrg / Language names' is visible, along with a 'translated 95%' indicator. A yellow banner states: 'Contribution to this translation requires you to agree with a contributor agreement.' with a 'View contributor agreement' button. Below this, a tabbed interface shows 'Translations' selected. A table lists translations for Czech, Hebrew, Hungarian, and English, all under the GPL-3.0 license. The Hungarian row shows 81% translated, 4 untranslatable words, and 5 untranslated words. At the bottom, there's a 'Start new translation' button and a footer with 'Powered by Weblate 4.6' and various links.

Language	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Czech 🇨🇪 GPL-3.0	✓					
Hebrew 🇮🇱 GPL-3.0	✓					
Hungarian 🇮🇪 GPL-3.0	81%	4	5			
English 🇬🇧 GPL-3.0	✓					

The entered text is formatted into paragraphs and external links can be included. HTML markup can not be used.

2.10.3 User licenses

Any user can review all translation licenses of all public projects on the instance from their profile:

The screenshot shows the Weblate user interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks' menus, along with icons for settings, a plus sign, a user profile, and a menu. Below this is a light blue bar with 'Your profile' and a list of tabs: 'Languages', 'Preferences', 'Notifications', 'Account', 'Profile', 'Licenses' (which is highlighted), 'Audit log', and 'API access'. The main content area has a header 'Licenses'. The first section contains a warning: 'Please pay attention to the licensing info, as this specifies how translations can be used. By registering you agree to use your name and e-mail in the commits, and provide your contribution under the license defined by each localization project. You have agreed to the following as a contributor:' followed by a bulleted list with 'WeblateOrg/Language names'. The second section is titled 'Licenses for individual translations' and lists 'GNU General Public License v3.0 or later' with a link to 'GPL-3.0' and 'MIT License' with a link to 'MIT'. Below these are links to various projects: 'WeblateOrg/WeblateOrg', 'WeblateOrg/Djangojs', 'WeblateOrg/Django', 'WeblateOrg/Language names', and 'WeblateOrg/Android'. At the bottom of the page is a footer: 'Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate'.

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* configuration 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*.

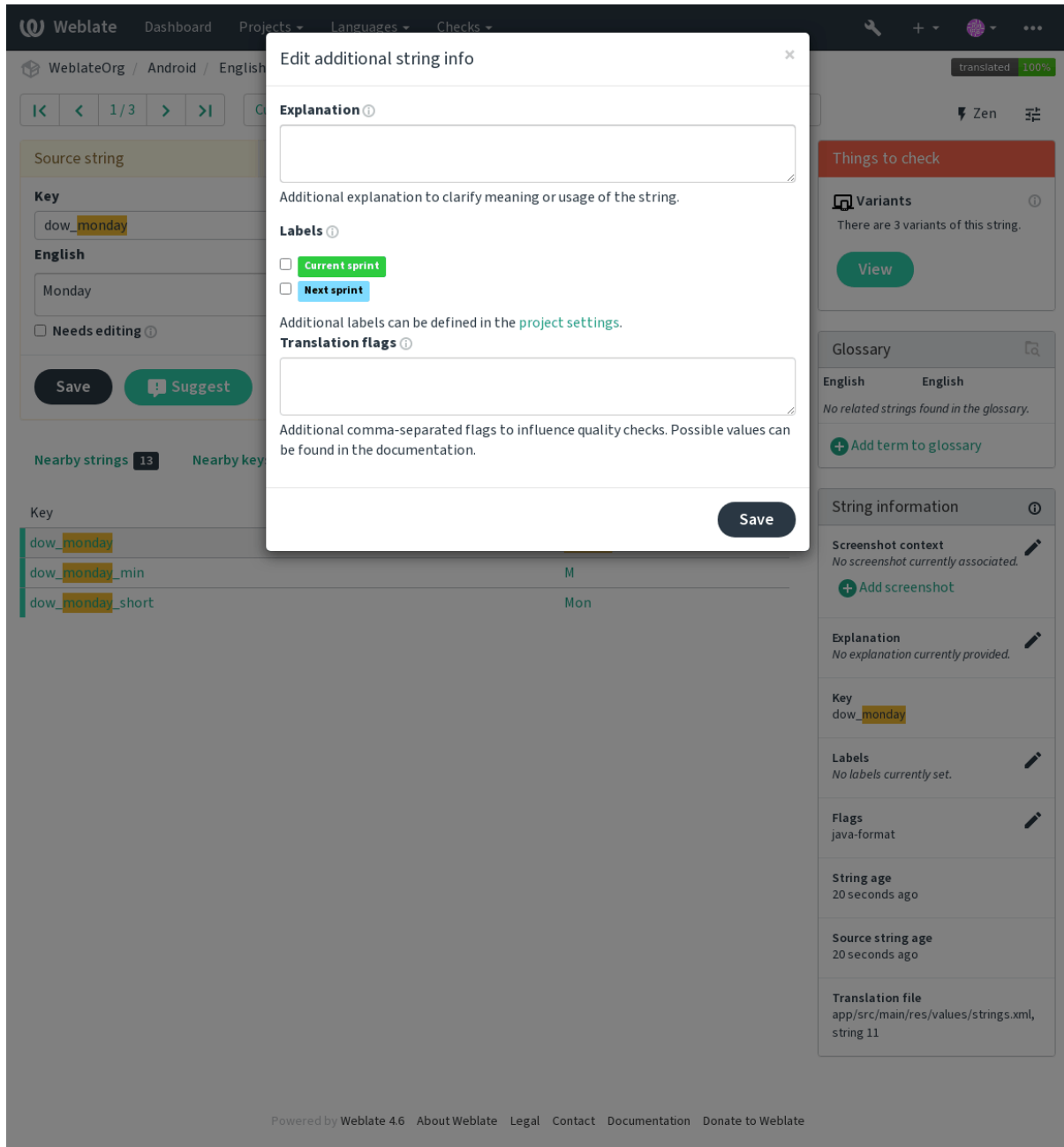
You can combine these with *access control* into one of the following setups:

- 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*.

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 ⓘ

Save Suggest Skip

Nearby strings 26 Comments Automatic suggestions Other languages 4 History

Context	English	Czech
	Files	Soubory
	Automatic translation	Automatický překlad
	Add new translation string	Add new translation string
	Translation status	Stav překladu
	%(count)s word	%(count)s slovo
	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 příklady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahráný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nepamenejte to povolit.
	The uploaded file will be merged with the current translation.	Nahráný soubor bude sloučen se stávajícími překlady.
	The fulltext search might not work properly as the fulltext index for this translation is not yet up to date.	Fulltextové vyhledávání nemusí fungovat správe, protože fulltextový index pro tento překlad ještě není plně zpracován.
	Review	Kontrola
	Review translations touched by other users.	Zkontrolovat překlady od ostatních uživatelů.
	Start review	Začít kontrolu
	Percent	Procenta
	Total	Celkem
	Failing check	Neúspěšných kontrol
	Last activity	Poslední aktivity
	Last change	Poslední změna
	Last author	Poslední autor
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

Source string age

3 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 11

Priorización de cadenas

Nuevo en la versión 2.0.

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

Nuevo en la versión 2.4.

Distinto en la versión 3.3: Previously called *Quality checks flags*, it no longer configures only checks.

The default set of translation flags is determined by the translation *Configuración de componentes* and the translation file. However, you might want to use it to customize this per source string.

Ver también:

[Comprobaciones de calidad](#)

Explicación

Distinto en la versión 4.1: In previous versions this has been called *Extra context*.

Utilice la explicación para aclarar el ámbito o la utilización de la traducción. Puede utilizar Markdown para incluir enlaces y dar formato.

Contexto visual para cadenas

Nuevo en la versión 2.9.

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:

Webate

Dashboard

Projects

Languages

Checks

WebateOrg / Django / Czech / Translate

translated 96%

<<

<

11 / 26

>

>>

All strings

Position and priority

Translation

Explanation

Help text for automatic translation tool

English

Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

Needs editing

Save

Suggest

Skip

Nearby strings

Comments

Automatic suggestions

Other languages

History

Translation memory

Translation	Source	Origin	Similarity		
Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Webate Translation Memory (Project: weblateorg/django) Webate Translation Memory (Shared: weblateorg/django) Webate (WeblateOrg/Django)	100 %	Copy	Copy and save

Glossary

English

Czech

machine translation

strojový překlad

project

projekt

Add term to glossary

String information

Screenshot context

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

14 seconds ago

Source string age

14 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 11

Powered by Weblate 4.6

About Weblate

Legal

Contact

Documentation

Donate to Weblate

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 to do so.

Once a screenshot is uploaded, this interface handles management and source string association:

260

Capítulo 2. Documentación para administradores

Webate

Dashboard

Projects

Languages

Checks









WebateOrg / Django / Screenshots / Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			
Screenshot is shown to add visual context for all listed source strings.			

Assign source strings

English	Location	Assigned screenshots	Actions
Loading results...			

Source string search

Search

Automatically recognize

Image

Source string

Hello, world! 

One
Orangutan has %d banana. 

Other
Orangutan has %d bananas. 

Try Weblate at <http://demo.weblate.org/>! 

Thank you for using Weblate.

Screenshot is shown to add visual context for all listed source strings.

Edit screenshot

Screenshot name

Automatic translation

Image

Currently: [screenshots/screenshot.png](#)

Change:

Choose File

No file chosen

Upload JPEG or PNG images up to 2000x2000 pixels.

Save

Screenshot details

Created	now
Uploaded by	 testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

2.12 Comprobaciones y correcciones

2.12.1 Correcciones automáticas personalizadas

You can also implement your own automatic fixup in addition to the standard ones and include them in `AUTO-FIX_LIST`.

Las correcciones automáticas son potentes pero pueden causar daños; tenga cuidado al escribir una.

For example, the following automatic fixup would replace every occurrence of the string `foo` in a translation with `bar`:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://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/>.
#

from django.utils.translation import gettext_lazy as _

from weblate.trans.autofixes.base import AutoFix

class ReplaceFooWithBar(AutoFix):
    """Replace foo with bar."""

    name = _("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False
```

To install custom checks, provide a fully-qualified path to the Python class in the `AUTOFIX_LIST`, see *Comprobaciones de calidad, complementos y correcciones automáticas personalizadas*.

2.12.2 Personalizar el comportamiento mediante indicadores

You can fine-tune the behavior of Weblate (mostly checks) for each source string (in source strings review, 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, the parameters are separated with colon. You can use quotes to include whitespace or special chars in the string. For example:

```
placeholders:"special:value":"other value", regex:.*
```

Here is a list of flags currently accepted:

rst-text Treat a text as an reStructuredText document, affects *Traducción no modificada*.

md-text Treat text as a Markdown document.

dos-eol Uses DOS end-of-line markers instead of Unix ones (\r\n instead of \n).

url The string should consist of only a URL.

safe-html The string should be HTML safe, see *HTML inseguro*.

read-only La cadena es de solo lectura y no debe editarse en Weblate; vea *Cadenas de solo lectura*.

priority:N Priority of the string. Higher priority strings are presented first for translation. The default priority is 100, the higher priority a string has, the earlier it is offered for translation.

max-length:N Limit the maximal length for a string to N characters, see *Longitud máxima de la traducción*.

xml-text Treat text as XML document, affects *Sintaxis XML* and *Marcación XML*.

font-family:NOMBRE Define font-family for rendering checks, see *Gestionar tipos de letra*.

font-weight:PESO Define font-weight for rendering checks, see *Gestionar tipos de letra*.

font-size:TAMAÑO Define font-size for rendering checks, see *Gestionar tipos de letra*.

font-spacing:ESPACIADO Define letter spacing for rendering checks, see *Gestionar tipos de letra*.

placeholders:NAME:NAME2:... Placeholder strings expected in translation, see *Sustitutorios*.

replacements:FROM:TO:FROM2:TO2... Replacements to perform when checking resulting text parameters (for example in *Tamaño máximo de la traducción* or *Longitud máxima de la traducción*). The typical use case for this is to expand placeables to ensure that the text fits even with long values, for example: `replacements:%s:"John Doe"`.

variants:SOURCE Mark this string as a variant of string with matching source. See variants.

regex:REGEX Expresión regular utilizada para relacionar traducciones; vea *Expresión regular*.

forbidden Indicates forbidden translation in a glossary, see *Traducciones prohibidas*.

python-format, c-format, php-format, python-brace-format, javascript-format, c-sharp-format, java-format Treats all strings like format strings, affects *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Cadenas formateadas*, *Traducción no modificada*.

strict-same Hace que «Traducción no modificada» evite la lista de palabras incorporada; vea *Traducción no modificada*.

check-glossary Enable the «Does not follow glossary» quality check.

ignore-bbcode Pasa por alto la comprobación de calidad «Marcación BBcode».

ignore-duplicate Pasa por alto la comprobación de calidad «Palabras consecutivas duplicadas».

ignore-check-glossary Pasa por alto la comprobación de calidad «No se ajusta al glosario».

ignore-double-space Pasa por alto la comprobación de calidad «Espacio duplicado».

ignore-angularjs-format Pasa por alto la comprobación de calidad «Cadena de interpolación de AngularJS».

ignore-c-format Pasa por alto la comprobación de calidad «Formato C».

ignore-c-sharp-format Pasa por alto la comprobación de calidad «Formato C#».

ignore-es-format Pasa por alto la comprobación de calidad «Literales de plantilla ECMAScript».

ignore-i18next-interpolation Pasa por alto la comprobación de calidad «Interpolación con i18next».

ignore-java-format Pasa por alto la comprobación de calidad «Formato Java».

ignore-java-messageformat Pasa por alto la comprobación de calidad «MessageFormat de Java».

ignore-javascript-format Pasa por alto la comprobación de calidad «Formato JavaScript».

ignore-lua-format Pasa por alto la comprobación de calidad «Formato Lua».

ignore-percent-placeholders Pasa por alto la comprobación de calidad «Sustitutorios con signo de porcentaje».

ignore-perl-format Pasa por alto la comprobación de calidad «Formato Perl».

ignore-php-format Pasa por alto la comprobación de calidad «Formato PHP».

ignore-python-brace-format Pasa por alto la comprobación de calidad «Formato de llaves de Python».

ignore-python-format Pasa por alto la comprobación de calidad «Formato Python».

ignore-qt-format Pasa por alto la comprobación de calidad «Formato Qt».

ignore-qt-plural-format Pasa por alto la comprobación de calidad «Formato de plurales de Qt».

ignore-ruby-format Pasa por alto la comprobación de calidad «Formato Ruby».

ignore-vue-format Skip the «Vue I18n formatting» quality check.

ignore-translated Pasa por alto la comprobación de calidad «Se había traducido».

ignore-inconsistent Pasa por alto la comprobación de calidad «Incoherente».

ignore-kashida Pasa por alto la comprobación de calidad «Kashida utilizado».

ignore-md-link Pasa por alto la comprobación de calidad «Enlaces de Markdown».

ignore-md-reflink Pasa por alto la comprobación de calidad «Referencias de Markdown».

ignore-md-syntax Pasa por alto la comprobación de calidad «Sintaxis de Markdown».

ignore-max-length Pasa por alto la comprobación de calidad «Longitud máxima de la traducción».

ignore-max-size Pasa por alto la comprobación de calidad «Tamaño máximo de la traducción».

ignore-escaped-newline Pasa por alto la comprobación de calidad «n desiguales».

ignore-end-colon Pasa por alto la comprobación de calidad «Dos puntos desiguales».

ignore-end-ellipsis Pasa por alto la comprobación de calidad «Puntos suspensivos desiguales».

ignore-end-exclamation Pasa por alto la comprobación de calidad «Signo de exclamación desigual».

ignore-end-stop Pasa por alto la comprobación de calidad «Punto final desigual».

ignore-end-question Pasa por alto la comprobación de calidad «Signo de interrogación desigual».

ignore-end-semicolon Pasa por alto la comprobación de calidad «Punto y coma desigual».

ignore-newline-count Pasa por alto la comprobación de calidad «Saltos de renglón desiguales».

ignore-plurals Pasa por alto la comprobación de calidad «Faltan plurales».

ignore-placeholders Pasa por alto la comprobación de calidad «Sustitutorios».

ignore-punctuation-spacing Pasa por alto la comprobación de calidad «Espaciado de puntuación».

ignore-regex Pasa por alto la comprobación de calidad «Expresión regular».

ignore-same-plurals Pasa por alto la comprobación de calidad «Mismos plurales».

ignore-begin-newline Pasa por alto la comprobación de calidad «Salto de renglón al inicio».

ignore-begin-space Pasa por alto la comprobación de calidad «Espacios iniciales».

ignore-end-newline Pasa por alto la comprobación de calidad «Salto de renglón al final».

ignore-end-space Pasa por alto la comprobación de calidad «Espacio al final».

ignore-same Pasa por alto la comprobación de calidad «Traducción no modificada».

ignore-safe-html Pasa por alto la comprobación de calidad «HTML inseguro».

ignore-url Pasa por alto la comprobación de calidad «URL».

ignore-xml-tags Pasa por alto la comprobación de calidad «Marcación XML».

ignore-xml-invalid Pasa por alto la comprobación de calidad «Sintaxis XML».

ignore-zero-width-space Pasa por alto la comprobación de calidad «Espacio de anchura cero».

ignore-ellipsis Pasa por alto la comprobación de calidad «Puntos suspensivos».

ignore-long-untranslated Pasa por alto la comprobación de calidad «Largamente no traducida».

ignore-multiple-failures Pasa por alto la comprobación de calidad «Varias comprobaciones fallidas».

ignore-unnamed-format Pasa por alto la comprobación de calidad «Varias variables sin nombre».

ignore-optional-plural Pasa por alto la comprobación de calidad «No pluralizada».

Nota: Generally the rule is named `ignore-*` for any check, using its identifier, so you can use this even for your custom checks.

These flags are understood both in *Configuración de componentes* settings, per source string settings and in the translation file itself (for example in GNU gettext).

2.12.3 Forzar comprobaciones

Nuevo en la versión 3.11.

You can configure a list of checks which can not be ignored by setting *Comprobaciones obligatorias* in *Configuración de componentes*. Each listed check can not be ignored in the user interface and any string failing this check is marked as *Needs editing* (see *Translation states*).

2.12.4 Gestionar tipos de letra

Nuevo en la versión 3.7.

Consejo: Fonts uploaded into Weblate are used purely for purposes of the *Tamaño máximo de la traducción* check, they do not have an effect in Weblate user interface.

The *Tamaño máximo de la traducción* check used to calculate dimensions of the rendered text needs font to be loaded into Weblate and selected using a translation flag (see *Personalizar el comportamiento mediante indicadores*).

Weblate font management tool in *Fonts* under the *Manage* menu of your translation project provides interface to upload and manage fonts. TrueType or OpenType fonts can be uploaded, set up font-groups and use those in the check.

The font-groups allow you to define different fonts for different languages, which is typically needed for non-latin languages:

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / Font groups / default-font

Font group

Name	default-font		
Default font	Source Sans Pro Bold		
Japanese	language override	Droid Sans Fallback Regular	Remove
Korean	language override	Droid Sans Fallback Regular	Remove

Delete

Add language override

Language

Font

Save

Edit font group

Font group name

default-font

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

Source Sans Pro Bold

Default font is used unless per language override matches.

Save

The font-groups are identified by name, which can not contain whitespace or special characters, so that it can be easily used in the check definition:

 Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

 WeblateOrg / Fonts

Font groups

Fonts

Group name	Default font	Language overrides	
default-font	Source Sans Pro Bold	Japanese: Droid Sans Fallback Regular Korean: Droid Sans Fallback Regular	

Add font group

Font group name

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

▼

Default font is used unless per language override matches.



Powered by Weblate 4.6

About Weblate

Legal

Contact

Documentation

Donate to Weblate

Font-family and style is automatically recognized after uploading them:

 Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

 WeblateOrg / Fonts / Droid Sans Fallback Regular

Font	
Font family	Droid Sans Fallback
Font style	Regular
File size	3939852
Created	now
Uploaded by	 testuser
Used in groups	
	

Powered by Weblate 4.6

About Weblate

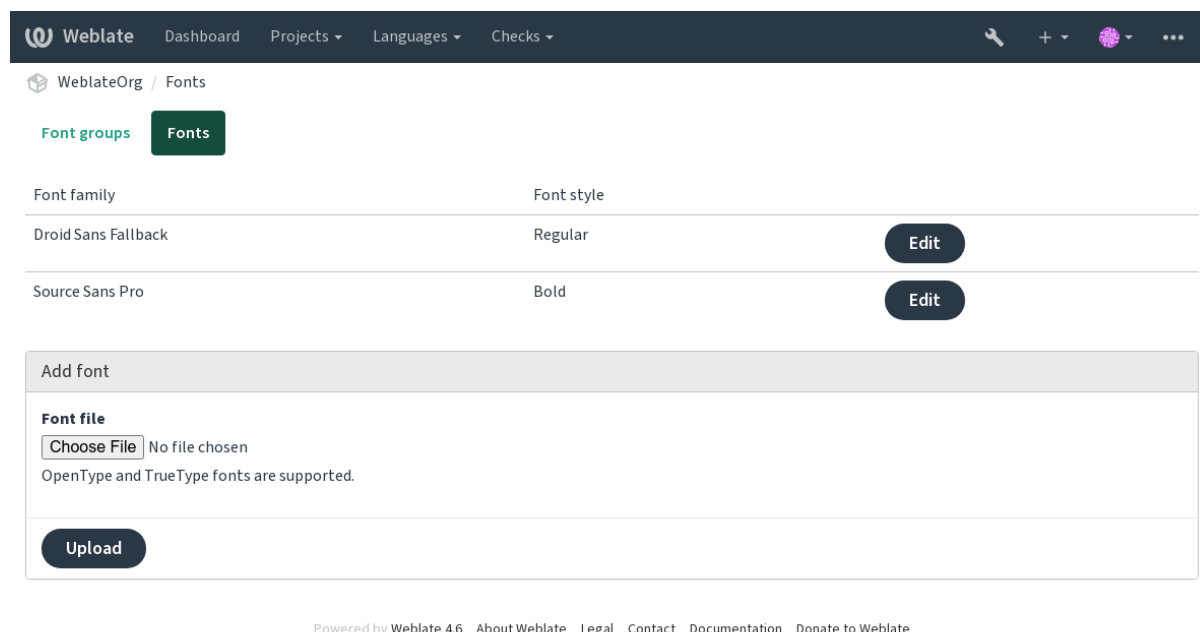
Legal

Contact

Documentation

Donate to Weblate

You can have a number of fonts loaded into Weblate:



To use the fonts for checking the string length, pass it the appropriate flags (see *Personalizar el comportamiento mediante indicadores*). You will probably need the following ones:

max-size:500 Define la anchura máxima.

font-family:ubuntu Defines font group to use by specifying its identifier.

font-size:22 Define el tamaño del texto.

2.12.5 Writing own checks

A wide range of quality checks are built-in, (see *Comprobaciones de calidad*), though they might not cover everything you want to check. The list of performed checks can be adjusted using `CHECK_LIST`, and you can also add custom checks.

1. Subclass the `weblate.checks.Check`
2. Defina algunos atributos.
3. Implement either the `check` (if you want to deal with plurals in your code) or the `check_single` method (which does it for you).

He aquí algunos ejemplos:

To install custom checks, provide a fully-qualified path to the Python class in the `CHECK_LIST`, see *Comprobaciones de calidad, complementos y correcciones automáticas personalizadas*.

Checking translation text does not contain «foo»

Esta es una comprobación bastante simple que se limita a revisar si a la traducción le falta la cadena «tal».

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://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
```

(continué en la próxima página)

(proviene de la página anterior)

```
# (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/>.
#
"""Simple quality check example."""

from django.utils.translation import gettext_lazy as _

from weblate.checks.base import TargetCheck

class FooCheck(TargetCheck):

    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_single(self, source, target, unit):
        return "foo" in target
```

Checking that Czech translation text plurals differ

Comprobación que utiliza la información de idioma para cerciorarse de que las dos formas de plural en checo no sean iguales.

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://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/>.
#
"""Quality check example for Czech plurals."""

from django.utils.translation import gettext_lazy as _
```

(continué en la próxima página)


```
from weblate.checks.base import TargetCheck

class PluralCzechCheck(TargetCheck):

    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_target_unit(self, sources, targets, unit):
        if self.is_language(unit, ("cs",)):
            return targets[1] == targets[2]
        return False

    def check_single(self, source, target, unit):
        """We don't check target strings here."""
        return False
```

2.13 Traducción automática

Built-in support for several machine translation services and can be turned on by the administrator using `MT_SERVICES` for each one. They come subject to their terms of use, so ensure you are allowed to use them how you want.

The source language can be configured at *Configuración de proyectos*.

2.13.1 amaGama

Special installation of `tmserver` run by the authors of Virtaal.

Turn on this service by adding `weblate.machinery.tmserver.AmagamaTranslation` to `MT_SERVICES`.

Ver también:

Installing amaGama, Amagama, amaGama Translation Memory

2.13.2 Apertium

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.

Turn on this service by adding `weblate.machinery.apertium.ApertiumAPYTranslation` to `MT_SERVICES` and set `MT_APERTIUM_APY`.

Ver también:

`MT_APERTIUM_APY`, Apertium website, Apertium APy documentation

2.13.3 AWS

Nuevo en la versión 3.1.

Amazon Translate is a neural machine translation service for translating text to and from English across a breadth of supported languages.

1. Turn on this service by adding `weblate.machinery.aws.AWSTranslation` to `MT_SERVICES`.
2. Install the *boto3* module.
3. Configure Weblate.

Ver también:

`MT_AWS_REGION`, `MT_AWS_ACCESS_KEY_ID`, `MT_AWS_SECRET_ACCESS_KEY` [Amazon Translate Documentation](#)

2.13.4 Baidu API machine translation

Nuevo en la versión 3.2.

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.

Turn on this service by adding `weblate.machinery.baidu.BaiduTranslation` to `MT_SERVICES` and set `MT_BAIDU_ID` and `MT_BAIDU_SECRET`.

Ver también:

`MT_BAIDU_ID`, `MT_BAIDU_SECRET` [Baidu Translate API](#)

2.13.5 DeepL

Nuevo en la versión 2.20.

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.

Turn on this service by adding `weblate.machinery.deepl.DeepLTranslation` to `MT_SERVICES` and set `MT_DEEPL_KEY`.

Consejo: In case you have subscription for CAT tools, you are supposed to use «v1 API» instead of default «v2» used by Weblate (it is not really an API version in this case). You can toggle this by `MT_DEEPL_API_VERSION`.

Ver también:

`MT_DEEPL_KEY`, `MT_DEEPL_API_VERSION`, [DeepL website](#), [DeepL pricing](#), [DeepL API documentation](#)

2.13.6 Glosbe

Free dictionary and translation memory for almost every living language.

The API is gratis to use, but subject to the used data source license. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Turn on this service by adding `weblate.machinery.glosbe.GlosbeTranslation` to `MT_SERVICES`.

Ver también:

[Glosbe website](#)

2.13.7 Google Translate

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.

To turn on this service, add `weblate.machinery.google.GoogleTranslation` to `MT_SERVICES` and set `MT_GOOGLE_KEY`.

Ver también:

`MT_GOOGLE_KEY`, [Google translate documentation](#)

2.13.8 Google Translate API V3 (Advanced)

Machine translation service provided by Google Cloud services.

This service differs from the former one in how it authenticates. To enable service, add `weblate.machinery.google.v3.GoogleV3Translation` to `MT_SERVICES` and set

- `MT_GOOGLE_CREDENTIALS`
- `MT_GOOGLE_PROJECT`

If `location` fails, you may also need to specify `MT_GOOGLE_LOCATION`.

Ver también:

`MT_GOOGLE_CREDENTIALS`, `MT_GOOGLE_PROJECT`, `MT_GOOGLE_LOCATION` [Google translate documentation](#)

2.13.9 Microsoft Cognitive Services Translator

Nuevo en la versión 2.10.

Machine translation service provided by Microsoft in Azure portal as a one of Cognitive Services.

Weblate implements Translator API V3.

To enable this service, add `weblate.machinery.microsoft.MicrosoftCognitiveTranslation` to `MT_SERVICES` and set `MT_MICROSOFT_COGNITIVE_KEY`.

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 `MT_MICROSOFT_REGION` to locale of your service.

Ver también:

`MT_MICROSOFT_COGNITIVE_KEY`, `MT_MICROSOFT_REGION`, [Cognitive Services - Text Translation API](#), [Microsoft Azure Portal](#)

2.13.10 Servicio terminológico de Microsoft

Nuevo en la versión 2.19.

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.

Turn this service on by adding `weblate.machinery.microsoftterminology.MicrosoftTerminologyService` to `MT_SERVICES`.

Ver también:

API del Servicio terminológico de Microsoft

2.13.11 ModernMT

Nuevo en la versión 4.2.

Turn this service on by adding `weblate.machinery.modernmt.ModernMTTranslation` to `MT_SERVICES` and configure `MT_MODERNMT_KEY`.

Ver también:

ModernMT API, `MT_MODERNMT_KEY`, `MT_MODERNMT_URL`

2.13.12 MyMemory

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 `MT_MYMEMORY_EMAIL`. You can also ask them for more.

Turn on this service by adding `weblate.machinery.mymemory.MyMemoryTranslation` to `MT_SERVICES` and set `MT_MYMEMORY_EMAIL`.

Ver también:

`MT_MYMEMORY_EMAIL`, `MT_MYMEMORY_USER`, `MT_MYMEMORY_KEY`, MyMemory website

2.13.13 Traducción automática por API de NetEase Sight

Nuevo en la versión 3.3.

Servicio de traducción automática brindado por NetEase.

Este servicio utiliza una API y debe adquirir una clave y secreto de NetEase.

Turn on this service by adding `weblate.machinery.youdao.NeteaseSightTranslation` to `MT_SERVICES` and set `MT_NETEASE_KEY` and `MT_NETEASE_SECRET`.

Ver también:

`MT_NETEASE_KEY`, `MT_NETEASE_SECRET` Netease Sight Translation Platform

2.13.14 tmserver

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:
2. Turn on this service by adding `weblate.machinery.tmserver.TMServerTranslation` to `MT_SERVICES`.

```
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
```

3. Inicie el tmserver para que escuche sus peticiones:

```
tmserver -d /var/lib/tm/db
```

4. Configure Weblate para que se comuniquen con este:

```
MT_TMSERVER = "http://localhost:8888/tmserver/"
```

Ver también:

[MT_TMSERVER](#), [tmserver](#) [Installing amaGama](#), [Amagama](#), [Amagama Translation Memory](#)

2.13.15 Yandex Translate

Servicio de traducción automática ofrecido por Yandex.

This service uses a Translation API, and you need to obtain an API key from Yandex.

Turn on this service by adding `weblate.machinery.yandex.YandexTranslation` to `MT_SERVICES`, and set `MT_YANDEX_KEY`.

Ver también:

[MT_YANDEX_KEY](#), [Yandex Translate API](#), [Powered by Yandex.Translate](#)

2.13.16 Youdao Zhiyun API machine translation

Nuevo en la versión 3.2.

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.

Turn on this service by adding `weblate.machinery.youdao.YoudaoTranslation` to `MT_SERVICES` and set `MT_YOUDAO_ID` and `MT_YOUDAO_SECRET`.

Ver también:

[MT_YOUDAO_ID](#), [MT_YOUDAO_SECRET](#) [Youdao Zhiyun Natural Language Translation Service](#)

2.13.17 Weblate

Weblate can be the source of machine translations as well. It is based on the Woosh fulltext engine, and provides both exact and inexact matches.

Turn on these services by adding `weblate.machinery.weblatetm.WeblateTranslation` to `MT_SERVICES`.

2.13.18 Weblate Translation Memory

Nuevo en la versión 2.20.

The *Memoria de traducción* can be used as a source for machine translation suggestions as well.

Turn on these services by adding `weblate.memory.machine.WeblateMemory` to the `MT_SERVICES`. This service is turned on by default.

2.13.19 SAP Translation Hub

Machine translation service provided by SAP.

You need to have a SAP account (and enabled the SAP Translation Hub in the SAP Cloud Platform) to use this service.

Turn on this service by adding `weblate.machinery.saptranslationhub.SAPTranslationHub` to `MT_SERVICES` and set the appropriate access to either sandbox or the productive API.

Nota: To access the Sandbox API, you need to set `MT_SAP_BASE_URL` and `MT_SAP_SANDBOX_APIKEY`.

To access the productive API, you need to set `MT_SAP_BASE_URL`, `MT_SAP_USERNAME` and `MT_SAP_PASSWORD`.

Ver también:

`MT_SAP_BASE_URL`, `MT_SAP_SANDBOX_APIKEY`, `MT_SAP_USERNAME`, `MT_SAP_PASSWORD`, `MT_SAP_USE_MT` SAP Translation Hub API

2.13.20 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 © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://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
```

(continué en la próxima página)

(proviene de la página anterior)

```
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation

class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

    def download_languages(self):
        """Return list of languages your machine translation supports."""
        return {"cs"}

    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```

You can list own class in `MT_SERVICES` and Weblate will start using that.

2.14 Complementos

Nuevo en la versión 2.19.

Addons provide ways to customize and automate the translation workflow. Admins can add and manage addons from the *Manage* ↓ *Addons* menu of each respective translation component.


Dashboard
Projects ▾
Languages ▾
Checks ▾
🔧
+
🌐
...


WebOrg / Language names / Addons

Installed addons

There are no addons currently installed.

Available addons


Automatic translation ⓘ

Automatically translates strings using machine translation or other components.

Install


Add missing languages ⓘ

Ensures a consistent set of languages is used for all components within a project.

project wide

Install


Component discovery ⓘ

Automatically adds or removes project components based on file changes in the version control system.

repository wide

Install


Bulk edit ⓘ

Bulk edit flags, labels, or states of strings.

Install


Statistics generator ⓘ

Generates a file containing detailed info about the translation status.

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 "Filemask") to match the POT file (as configured by "Template for new translations") using msgmerge.

Install


Squash Git commits ⓘ

Squash Git commits prior to pushing changes.

repository wide

Install


Stale comment removal ⓘ

Set a timeframe for removal of comments.

project wide

Install


Stale suggestion removal ⓘ

Set a timeframe for removal of suggestions.

project wide

Install

Some addons will ask for additional configuration during installation.

2.14.1 Complementos incorporados

Traducción automática

Nuevo en la versión 3.9.

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.

Publica las traducciones en una red de entrega de contenidos para su uso al regionalizar JavaScript o HTML.

Can be used to localize static HTML pages, or to load localization in the JavaScript code.

Generates a unique URL for your component you can include in HTML pages to localize them. See weblate-cdn for more details.

Ver también:

cdn-addon-config, weblate-cdn, cdn-addon-extract, cdn-addon-html

Quitar cadenas vacías

Nuevo en la versión 4.4.

Quita las cadenas que carecen de traducción de los archivos de traducción.

Use this to not have any empty strings in translation files (for example if your localization library displays them as missing instead of falling back to the source string).

Ver también:

Does Weblate update translation files besides translations?

Limpieza de archivos de traducción

Actualice todos los archivos de traducción para que coincidan con el archivo base monolingüe. Para la mayoría de los formatos de archivo, esto significa quitar las claves de traducción obsoletas que ya no están presentes en el archivo base.

Ver también:

Does Weblate update translation files besides translations?

Add missing languages

Ensures a consistent set of languages is used for all components within a project.

Missing languages are checked once every 24 hours, and when new languages are added in Weblate.

A diferencia de la mayoría de los demás, este complemento afecta a todo el proyecto.

Consejo: Auto-translate the newly added strings with *Traducción automática*.

Detección de componentes

Añade o quita automáticamente componentes del proyecto en función de los cambios de los archivos en el sistema de control de versiones.

Triggered each time the VCS is updated, and otherwise similar to the *import_project* management command. This way you can track multiple translation components within one VCS.

The matching is done using regular expressions enabling complex configuration, but some knowledge is required to do so. Some examples for common use cases can be found in the addon help section.

Once you hit *Save*, a preview of matching components will be presented, from where you can check whether the configuration actually matches your needs:

Weblate

Dashboard

Projects

Languages

Checks

+

WebOrg

Language names

Addons

Component discovery

Configure addon

Please review and confirm the matched components.

Component

Matched files

Following components would be created

Djangojs

weblate/locale/cs/LC_MESSAGES/djangojs.po (cs)

weblate/locale/hu/LC_MESSAGES/djangojs.po (hu)

weblate/locale/he/LC_MESSAGES/djangojs.po (he)

Django

weblate/locale/cs/LC_MESSAGES/django.po (cs)

weblate/locale/hu/LC_MESSAGES/django.po (hu)

weblate/locale/he/LC_MESSAGES/django.po (he)

☐ I confirm the above matches look correct

Regular expression to match translation files against

weblate/locale/(?P<language>[^\.]*)/LC_MESSAGES/(?P<component>[^\.]*).po

File format

gettext PO file

Customize the component name

{{ component|title }}

Define the monolingual base filename

Leave empty for bilingual translation files.

Define the base file for new translations

weblate/locale/{{ component }}.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Language filter

^(cs|he|hu)\$

Regular expression to filter translation files against when scanning for filemask.

☒ Clone addons from the main component to the newly created ones

☐ Remove components for inexistant files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description
(?P<language>[^\.]*)/(?P<component>[^\.]*).po	cs/application.po cs/website.po de/application.po de/website.po	One folder per language containing translation files for components.
locale/(?P<language>[^\.]*)/LC_MESSAGES/(?P<component>[^\.]*).po	locale/cs/LC_MESSAGES/application.po locale/cs/LC_MESSAGES/website.po locale/de/LC_MESSAGES/application.po locale/de/LC_MESSAGES/website.po	Usual structure for storing gettext PO files.
src/locale/(?P<component>[^\.]*).(?P<language>[^\.]*).po	src/locale/application.cs.po src/locale/website.cs.po src/locale/application.de.po src/locale/website.de.po	Using both component and language name within filename.
locale/(?P<language>[^\.]*)/(?P<component>[^\.]*)(?P=language).po	locale/cs/application/cs.po locale/cs/website/cs.po locale/de/application/de.po locale/de/website/de.po	Using language in both path and filename.
res/values-(?P<language>[^\.]*)/strings-(?P<component>[^\.]*).xml	res/values-cs/strings-about.xml res/values-cs/strings-help.xml res/values-de/strings-about.xml res/values-de/strings-help.xml	Android resource strings, split into several files.

You can use Django template markup in both component name and the monolingual base filename, for example:

{{ component }}

Component filename match

{{ component|title }}

Component filename with upper case first letter

Save

Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate

280

Capítulo 2. Documentación para administradores

Consejo: Component discovery addon uses *URL internos de Weblate*. It's a convenient way to share VCS setup between multiple components. Linked components use the local repository of the main component set up by filling `weblate://project/main-component` into the *Repositorio de código fuente* field (in *Manage* ↓ *Settings* ↓ *Version control system*) of each respective component. This saves time with configuration and system resources too.

Ver también:

Marcación de plantilla

Edición en masa

Nuevo en la versión 3.11.

Edite los indicadores, las etiquetas o los estados de las cadenas de manera masiva.

Automate labeling by starting out with the search query `NOT has:label` and add labels till all strings have all required labels. Other automated operations for Weblate metadata can also be done.

Ejemplos:

Tabla 4: Etiquetar cadenas nuevas automáticamente

Consulta de búsqueda	<code>NOT has:label</code>
Etiquetas que añadir	<code>recent</code>

Tabla 5: Marking all Archivos de metadatos de tiendas de aplicaciones changelog entries read-only

Consulta de búsqueda	<code>language:en AND key:changelogs/</code>
Indicadores de traducción que añadir	<code>read-only</code>

Ver también:

Edición en masa, *Personalizar el comportamiento mediante indicadores*, *labels*

Marcar traducciones sin cambios como «Necesitan edición»

Nuevo en la versión 3.1.

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: You might also want to tighten the *Traducción no modificada* check by adding `strict-same` flag to *Indicadores de traducción*.

Ver también:

Translation states

Marcar cadenas de origen nuevas como «Necesitan edición»

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:

Translation states

Marcar traducciones nuevas como «Necesitan edición»

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:

Translation states

Generador de estadísticas

Genera un archivo con información detallada sobre el estado de la traducción.

You can use a Django template in both filename and content, see *Marcación de plantilla* for a detailed markup description.

For example generating a summary file for each translation:

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

Generación de pseudorregiones

Genera una traducción mediante la adición automática de un prefijo y un sufijo a las cadenas de origen.

Pseudolocales are useful to find strings that are not prepared for localization. This is done by altering all translatable source strings to make it easy to spot unaltered strings when running the application in the pseudolocale language.

Finding strings whose localized counterparts might not fit the layout is also possible.

Consejo: You can use real languages for testing, but there are dedicated pseudolocales available in Weblate - *en_XA* and *ar_XB*.

Contribuidores en comentarios

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@cihar.com>, 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»

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

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

Actualiza el archivo LINGUAS al añadir una traducción nueva.

Generar archivos MO

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*.

Actualizar archivos PO para que coincidan con POT (msgmerge)

Actualiza todos los archivos PO (tal como se configuraron mediante *File mask*) a juego con el archivo POT (según esté configurado por *Plantilla para traducciones nuevas*) utilizando **msgmerge**.

Triggered whenever new changes are pulled from the upstream repository. Most msgmerge command-line options can be set up through the addon configuration.

Ver también:

Does Weblate update translation files besides translations?

Concentrar consignas de Git

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:

Nuevo en la versión 3.4.

- Todas las consignas en una sola
- Por idioma
- Por archivo

Nuevo en la versión 3.5.

- 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.

Nuevo en la versión 4.1.

Los mensajes de consigna originales pueden alterarse opcionalmente por un mensaje 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

Permite ajustar el comportamiento de salida de JSON, por ejemplo, sangría o clasificación.

Formatea el archivo de propiedades de Java

Ordena el archivo de propiedades de Java.

Retiro de comentarios obsoletos

Nuevo en la versión 3.7.

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.

Fijar un plazo para el borrado de sugerencias.

Can be very useful in connection with suggestion voting (see [Revisión por pares](#)) to remove suggestions which don't receive enough positive votes in a given timeframe.

Actualizar archivos RESX

Nuevo en la versión 3.9.

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* if you only want to remove stale translation keys.

Ver también:

Does Weblate update translation files besides translations?

Personalizar salida de YAML

Nuevo en la versión 3.10.2.

Permite ajustar el comportamiento de salida de YAML, como la longitud de los renglones y los saltos.

2.14.2 Personalizar la lista de complementos

The list of addons is configured by `WEBLATE_ADDONS`. To add another addon, simply include the absolute class name in this setting.

2.14.3 Escribir un complemento

You can write your own addons too, create a subclass of `weblate.addons.base.BaseAddon` to define the addon metadata, and then implement a callback to do the processing.

Ver también:

Desarrollo de complementos

2.14.4 Ejecutar secuencias de órdenes de un complemento

Addons can also be used to execute external scripts. This used to be integrated in Weblate, but now you have to write some code to wrap your script with an addon.

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://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/>.
#
"""Example pre commit script."""
```

(continué en la próxima página)

(proviene de la página anterior)

```
from django.utils.translation import gettext_lazy as _

from weblate.addons.events import EVENT_PRE_COMMIT
from weblate.addons.scripts import BaseScriptAddon


class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
    name = "weblate.example.pre"
    # Verbose name and long description
    verbose = _("Execute script before commit")
    description = _("This addon 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 *Comprobaciones de calidad, complementos y correcciones automáticas personalizadas*.

The script is executed with the current directory set to the root of the VCS repository for any given component.

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

Nuevo en la versión 2.11.

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

Nuevo en la versión 2.14.

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

Previous HEAD after update (only available after running the post-update hook).

WL_COMPONENT_SLUG

Nuevo en la versión 3.9.

Component slug used to construct URL.

WL_PROJECT_SLUG

Nuevo en la versión 3.9.

Project slug used to construct 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

Post-update repository processing

Can be used to update translation files when the VCS upstream source changes. To achieve this, please remember Weblate only sees files committed to the VCS, so you need to commit changes as a part of the 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

Use the commit script to automatically change a translation before it is committed to the repository.

It is passed as a single parameter consisting of the filename of a current translation.

2.15 Memoria de traducción

Nuevo en la versión 2.20.

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* addon.

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 `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.

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](#)

testuser / Translation memory

Translation memory status ⓘ

Number of your entries	0	Download as JSON	Download as TMX	Delete
Total number of entries	0			

Import translation memory

File

Choose File No file chosen

You can upload a TMX or JSON file.

Upload

Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate

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):

`dump_memory` Exporta la memoria a JSON

`import_memory` Importa archivos TMX o JSON en la memoria de traducción

2.16 Configuración

All settings are stored in `settings.py` (as is usual for Django).

Nota: After changing any of these settings, you need to restart Weblate - both WSGI and Celery processes.

In case it is run as `mod_wsgi`, you need to restart Apache to reload the configuration.

Ver también:

Please also check [Django's documentation](#) for parameters configuring Django itself.

2.16.1 AKISMET_API_KEY

Weblate can use Akismet to check incoming anonymous suggestions for spam. Visit akismet.com to purchase an API key and associate it with a site.

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

Nuevo en la versión 2.14.

Maximum number of failed authentication attempts before rate limiting is applied.

This is currently applied in the following locations:

- Logins. Deletes the account password, preventing the user from signing in without requesting a new password.
- Password resets. 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: The original on/off option was changed to differentiate which strings are accepted.

Actualiza todos los repositorios diariamente.

Consejo: Useful if you are not using *Actuadores de notificación* to update Weblate repositories automatically.

Nota: On/off options exist in addition to string selection for backward compatibility.

Las opciones son:

"none" No daily updates.

"remote" also False Actualiza los remotos solamente.

"full" also True Actualiza los remotos y fusiona la copia de trabajo.

Nota: This requires that *Tareas en segundo plano con Celery* is working, and will take effect after it is restarted.

2.16.6 AVATAR_URL_PREFIX

Prefix for constructing avatar URLs as: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. The following services are known to work:

Gravatar (default), as per <https://gravatar.com/> `AVATAR_URL_PREFIX` = `'https://www.gravatar.com/'`

Libravatar, as per <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

Nuevo en la versión 2.14.

How long the authentication token and temporary password from password reset e-mails is valid for. Set in number of seconds, defaulting to 172800 (2 days).

2.16.8 AUTH_PASSWORD_DAYS

Nuevo en la versión 2.15.

Por cuántos días debe permitirse el uso de la misma contraseña.

Nota: Password changes made prior to Weblate 2.15 will not be accounted for in this policy.

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 implementing the autofixer interface.

Correcciones disponibles:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace` Matches whitespace at the start and end of the string to the source.

`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.html.BleachHTML` Removes unsafe HTML markup from strings flagged as `safe-html` (see *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.

Defines how often lengthy maintenance tasks should be triggered for a component.

Right now this controls:

- *Traducción automática* addon
- *Comprobaciones y correcciones* recalculation

Elecciones posibles:

- monthly (valor predeterminado)
- weekly
- daily
- never

Nota: No es recomendable aumentar la frecuencia cuando Weblate contiene miles de componentes.

2.16.11 BASE_DIR

Base directory where Weblate sources are located. Used to derive several other paths by default:

- *DATA_DIR*

Default value: Top level directory of Weblate sources.

2.16.12 BASIC_LANGUAGES

Nuevo en la versión 4.4.

List of languages to offer users for starting new translation. When not specified 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. The project admins are still presented with full selection of languages defined in Weblate.

Nota: Esto no define idiomas nuevos en Weblate, sino que se limita a filtrar los que ya existen en la base de datos.

Ejemplo:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Ver también:

Language definitions

2.16.13 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Customize Content-Security-Policy header for Weblate. The header is automatically generated based on enabled integrations with third-party services (Matomo, Google Analytics, Sentry, ...).

All these default to empty list.

Ejemplo:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Ver también:

Content security policy, Directiva de seguridad de contenido (CSP)

2.16.14 CHECK_LIST

Lista de comprobaciones de calidad que ejecutar en una traducción.

Nota: Provide a fully-qualified path to the Python class implementing the check interface.

Adjust the list of checks to include ones relevant to you.

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 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 *updatechecks*.

Ver también:

Comprobaciones de calidad, *Personalizar el comportamiento mediante indicadores*

2.16.15 COMMENT_CLEANUP_DAYS

Nuevo en la versión 3.6.

Delete comments after a given number of days. Defaults to `None`, meaning no deletion at all.

2.16.16 COMMIT_PENDING_HOURS

Nuevo en la versión 2.10.

Number of hours between committing pending changes by way of the background task.

Ver también:

Configuración de componentes, Antigüedad de cambios por consignar, Efectuar tareas de mantenimiento, `commit_pending`

2.16.17 CONTACT_FORM

Nuevo en la versión 4.6.

Configures how e-mail from the contact form is being sent. Choose a configuration that matches your mail server configuration.

"reply-to" The sender is used in as *Reply-To*, this is the default behaviour.

"from" The sender is used in as *From*. Your mail server needs to allow sending such e-mails.

2.16.18 DATA_DIR

The folder Weblate stores all data in. It contains links to VCS repositories, a fulltext index and various configuration files for external tools.

A menudo existen los subdirectorios siguientes:

home Directorio de inicio utilizado para invocar secuencias de órdenes.

ssh Claves SSH y su configuración.

static Default location for static Django files, specified by `STATIC_ROOT`. See *Serving static files*.

media Default location for Django media files, specified by `MEDIA_ROOT`. Contains uploaded screenshots.

vcs Version control repositories for translations.

backups Daily backup data, please check *Datos volcados para las copias de respaldo* for details.

celery Celery scheduler data, see *Tareas en segundo plano con Celery*.

fonts: User-uploaded fonts, see *Gestionar tipos de letra*.

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
```

Defaults to `$BASE_DIR/data`.

Ver también:

BASE_DIR, Permisos del sistema de archivos, Respaldo y trasladar Weblate

2.16.19 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

2.16.20 DEFAULT_ACCESS_CONTROL

Nuevo en la versión 3.3.

La configuración de control de acceso predeterminada para proyectos nuevos:

0 *Público*

1 *Protegido*

100 *Privado*

200 *Personalizado*

Use *Custom* if you are managing ACL manually, which means not relying on the internal Weblate management.

Ver también:

Control de acceso al proyecto, Control de acceso

2.16.21 DEFAULT_AUTO_WATCH

Nuevo en la versión 4.5.

Configures whether *Automatically watch projects on contribution* should be turned on for new users. Defaults to `True`.

Ver también:

Notificaciones

2.16.22 DEFAULT_RESTRICTED_COMPONENT

Nuevo en la versión 4.1.

El valor predeterminado para la restricción de componentes.

Ver también:

Acceso restringido, Scope of groups

2.16.23 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DE- FAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DE- FAULT_MERGE_MESSAGE

Default commit messages for different operations, please check *Configuración de componentes* for details.

Ver también:

Marcación de plantilla, Configuración de componentes, Mensajes de consigna, de adición, de eliminación, de fusión y de complemento

2.16.24 DEFAULT_ADDONS

Los complementos predeterminados que se instalarán en todos los componentes que se creen.

Nota: Esta configuración solo surte efecto en los componentes nuevos.

Ejemplo:

```
DEFAULT_ADDONS = {  
    # Addon with no parameters  
    "weblate.flags.target_edit": {},  
    # Addon with parameters  
    "weblate.autotranslate.autotranslate": {  
        "mode": "suggest",  
        "filter_type": "todo",  
        "auto_source": "mt",  
        "component": "",  
        "engines": ["weblate-translation-memory"],  
        "threshold": "80",  
    },  
}
```

Ver también:

install_addon, WEBLATE_ADDONS

2.16.25 DEFAULT_COMMITER_EMAIL

Nuevo en la versión 2.4.

Committer e-mail address defaulting to `noreply@weblate.org`.

Ver también:

DEFAULT_COMMITER_NAME

2.16.26 DEFAULT_COMMITER_NAME

Nuevo en la versión 2.4.

El nombre del consignante, cuyo valor predeterminado es `Weblate`.

Ver también:

DEFAULT_COMMITER_EMAIL

2.16.27 DEFAULT_LANGUAGE

Nuevo en la versión 4.3.2.

Default source language to use for example in *Idioma del código fuente*.

Defaults to *en*. The matching language object needs to exist in the database.

Ver también:

Language definitions, Idioma del código fuente

2.16.28 DEFAULT_MERGE_STYLE

Nuevo en la versión 3.4.

Estilo de fusión para cualesquier componentes nuevos.

- *rebase* - default
- *merge*

Ver también:

Configuración de componentes, Estilo de fusión

2.16.29 DEFAULT_SHARED_TM

Nuevo en la versión 3.2.

Configures default value of *Utilizar memoria de traducción compartida* and *Contribuir a la memoria de traducción compartida*.

2.16.30 DEFAULT_TRANSLATION_PROPAGATION

Nuevo en la versión 2.5.

Default setting for translation propagation, defaults to `True`.

Ver también:

Configuración de componentes, Permitir propagación de traducciones

2.16.31 DEFAULT_PULL_MESSAGE

Title for new pull requests, defaulting to `'Update from Weblate'`.

2.16.32 ENABLE_AVATARS

Whether to turn on Gravatar-based avatars for users. By default this is on.

Avatars are fetched and cached on the server, lowering the risk of leaking private info, speeding up the user experience.

Ver también:

Caché de avatars, AVATAR_URL_PREFIX, Avatars

2.16.33 ENABLE_HOOKS

Whether to enable anonymous remote hooks.

Ver también:

Actuadores de notificación

2.16.34 ENABLE_HTTPS

Whether to send links to Weblate as HTTPS or HTTP. This setting affects sent e-mails and generated absolute URLs.

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 HTTPS URL.

The HTTPS redirection might be problematic in some cases and you might hit issue with infinite redirection in case you are using a reverse proxy doing 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*

2.16.35 ENABLE_SHARING

Turn on/off the *Share* menu so users can share translation progress on social networks.

2.16.36 GET_HELP_URL

Nuevo en la versión 4.5.2.

URL where support for your Weblate instance can be found.

2.16.37 GITLAB_CREDENTIALS

Nuevo en la versión 4.3.

List for credentials for GitLab servers.

Consejo: Use this in case you want Weblate to interact with more of them, for single GitLab endpoint stick with `GITLAB_USERNAME` and `GITLAB_TOKEN`.

```
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitlab.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.38 GITLAB_USERNAME

GitLab username used to send merge requests for translation updates.

Ver también:

GITLAB_CREDENTIALS, *GitLab*

2.16.39 GITLAB_TOKEN

Nuevo en la versión 4.3.

Ficha de acceso personal de GitLab utilizada para realizar llamadas de API para obtener actualizaciones de las traducciones.

Ver también:

GITLAB_CREDENTIALS, *GitLab*, GitLab: Personal access token

2.16.40 GITHUB_CREDENTIALS

Nuevo en la versión 4.3.

List for credentials for GitHub servers.

Consejo: Use this in case you want Weblate to interact with more of them, for single GitHub endpoint stick with *GITHUB_USERNAME* and *GITHUB_TOKEN*.

```
GITHUB_CREDENTIALS = {
  "api.github.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "github.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

2.16.41 GITHUB_USERNAME

GitHub username used to send pull requests for translation updates.

Ver también:

GITHUB_CREDENTIALS, *GitHub*

2.16.42 GITHUB_TOKEN

Nuevo en la versión 4.3.

GitHub personal access token used to make API calls to send pull requests for translation updates.

Ver también:

GITHUB_CREDENTIALS, *GitHub*, Creating a personal access token

2.16.43 GOOGLE_ANALYTICS_ID

Google Analytics ID to turn on monitoring of Weblate using Google Analytics.

2.16.44 HIDE_REPO_CREDENTIALS

Hide repository credentials from the web interface. In case you have repository URL with user and password, Weblate will hide it when related info is shown to users.

For example instead of `https://user:password@git.example.com/repo.git` it will show just `https://git.example.com/repo.git`. It tries to clean up VCS error messages too in a similar manner.

Nota: This is turned on by default.

2.16.45 HIDE_VERSION

Nuevo en la versión 4.3.1.

Hides version information from unauthenticated users. This also makes all documentation links point to latest version instead of the documentation matching currently installed version.

Hiding version is recommended security practice in some corporations, but it doesn't prevent attacker to figure out version by probing the behavior.

Nota: This is turned off by default.

2.16.46 IP_BEHIND_REVERSE_PROXY

Nuevo en la versión 2.14.

Indicates whether Weblate is running behind a reverse proxy.

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: This is not on by default.

Ver también:

Running behind reverse proxy, Rate limiting, IP_PROXY_HEADER, IP_PROXY_OFFSET

2.16.47 IP_PROXY_HEADER

Nuevo en la versión 2.14.

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`

2.16.48 IP_PROXY_OFFSET

Nuevo en la versión 2.14.

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 IP address.

Defaults to 0.

Ver también:

Running behind reverse proxy, *Rate limiting*, `SECURE_PROXY_SSL_HEADER`, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`

2.16.49 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/"
```

2.16.50 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",  
    ),  
]
```

2.16.51 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

2.16.52 LICENSE_REQUIRED

Defines whether the license attribute in *Configuración de componentes* is required.

Nota: This is off by default.

2.16.53 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`.

2.16.54 LOCALIZE_CDN_URL y LOCALIZE_CDN_PATH

These settings configure the *CDN de regionalización de JavaScript* addon. *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

2.16.55 LOGIN_REQUIRED_URLS

A list of URLs you want to require logging into. (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

2.16.56 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
)
```

2.16.57 MATOMO_SITE_ID

Identificador de un sitio en Matomo (antes Piwik) al que quiere realizar seguimiento.

Nota: Esta integración no admite el gestor de etiquetas de Matomo.

Ver también:

MATOMO_URL

2.16.58 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

2.16.59 MT_SERVICES

Distinto en la versión 3.0: The setting was renamed from MACHINE_TRANSLATION_SERVICES to MT_SERVICES to be consistent with other machine translation settings.

List of enabled machine translation services to use.

Nota: Many of the services need additional configuration like API keys, please check their documentation *Traducción automática* for more details.

```
MT_SERVICES = (
    "weblate.machinery.apertium.ApertiumAPYTranslation",
    "weblate.machinery.deepl.DeepLTranslation",
    "weblate.machinery.glosbe.GlosbeTranslation",
    "weblate.machinery.google.GoogleTranslation",
    "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    "weblate.machinery.mymemory.MyMemoryTranslation",
    "weblate.machinery.tmserver.AmagamaTranslation",
    "weblate.machinery.tmserver.TMServerTranslation",
    "weblate.machinery.yandex.YandexTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.machinery.saptranslationhub.SAPTranslationHub",
    "weblate.memory.machine.WeblateMemory",
)
```

Ver también:

Traducción automática, Sugerencias automáticas

2.16.60 MT_APERTIUM_APY

URL del servidor Apertium-APy, <https://wiki.apertium.org/wiki/Apertium-apy>

Ver también:

Apertium, Traducción automática, Sugerencias automáticas

2.16.61 MT_AWS_ACCESS_KEY_ID

Access key ID for Amazon Translate.

Ver también:

AWS, Traducción automática, Sugerencias automáticas

2.16.62 MT_AWS_SECRET_ACCESS_KEY

API secret key for Amazon Translate.

Ver también:

AWS, Traducción automática, Sugerencias automáticas

2.16.63 MT_AWS_REGION

Region name to use for Amazon Translate.

Ver también:

AWS, Traducción automática, Sugerencias automáticas

2.16.64 MT_Baidu_ID

Client ID for the Baidu Zhiyun API, you can register at <https://api.fanyi.baidu.com/api/trans/product/index>

Ver también:

Baidu API machine translation, Traducción automática, Sugerencias automáticas

2.16.65 MT_Baidu_SECRET

Client secret for the Baidu Zhiyun API, you can register at <https://api.fanyi.baidu.com/api/trans/product/index>

Ver también:

Baidu API machine translation, Traducción automática, Sugerencias automáticas

2.16.66 MT_DEEPL_API_VERSION

Nuevo en la versión 4.1.1.

API version to use with DeepL service. The version limits scope of usage:

v1 Is meant for CAT tools and is usable with user-based subscription.

v2 Is meant for API usage and the subscription is usage based.

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.

Ver también:

DeepL, Traducción automática, Sugerencias automáticas

2.16.67 MT_DEEPL_KEY

API key for the DeepL API, you can register at <https://www.deepl.com/pro.html>

Ver también:

DeepL, Traducción automática, Sugerencias automáticas

2.16.68 MT_GOOGLE_KEY

API key for Google Translate API v2, you can register at <https://cloud.google.com/translate/docs>

Ver también:

Google Translate, Traducción automática, Sugerencias automáticas

2.16.69 MT_GOOGLE_CREDENTIALS

API v3 JSON credentials file obtained in the Google cloud console. Please provide a full OS path. Credentials are per service-account affiliated with certain project. Please check <https://cloud.google.com/docs/authentication/getting-started> for more details.

2.16.70 MT_GOOGLE_PROJECT

Google Cloud API v3 project id with activated translation service and billing activated. Please check <https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project> for more details

2.16.71 MT_GOOGLE_LOCATION

API v3 Google Cloud App Engine may be specific to a location. Change accordingly if the default `global` fallback does not work for you.

Please check <https://cloud.google.com/appengine/docs/locations> for more details

Ver también:

Google Translate API V3 (Advanced)

2.16.72 MT_MICROSOFT_BASE_URL

Region base URL domain as defined in the «Base URLs» section.

Defaults to `api.cognitive.microsofttranslator.com` for Azure Global.

For Azure China, please use `api.translator.azure.cn`.

2.16.73 MT_MICROSOFT_COGNITIVE_KEY

Client key for the Microsoft Cognitive Services Translator API.

Ver también:

Microsoft Cognitive Services Translator, Traducción automática, Sugerencias automáticas, Cognitive Services - Text Translation API, Microsoft Azure Portal

2.16.74 MT_MICROSOFT_REGION

Region prefix as defined in the «Authenticating with a Multi-service resource» section.

2.16.75 MT_MICROSOFT_ENDPOINT_URL

Region endpoint URL domain for access token as defined in the «Authenticating with an access token» section.

Defaults to `api.cognitive.microsoft.com` for Azure Global.

For Azure China, please use your endpoint from the Azure Portal.

2.16.76 MT_MODERNMT_KEY

Clave de API para el motor de traducciones automáticas ModernMT.

Ver también:

ModernMT MT_MODERNMT_URL

2.16.77 MT_MODERNMT_URL

URL of ModernMT. It defaults to `https://api.modernmt.com/` for the cloud service.

Ver también:

ModernMT MT_MODERNMT_KEY

2.16.78 MT_MYMEMORY_EMAIL

MyMemory identification e-mail address. It permits 1000 requests per day.

Ver también:

MyMemory, Traducción automática, Sugerencias automáticas, MyMemory: API technical specifications

2.16.79 MT_MYMEMORY_KEY

MyMemory access key for private translation memory, use it with `MT_MYMEMORY_USER`.

Ver también:

MyMemory, Traducción automática, Sugerencias automáticas, MyMemory: API key generator

2.16.80 MT_MYMEMORY_USER

MyMemory user ID for private translation memory, use it with `MT_MYMEMORY_KEY`.

Ver también:

MyMemory, Traducción automática, Sugerencias automáticas, MyMemory: API key generator

2.16.81 MT_NETEASE_KEY

App key for NetEase Sight API, you can register at <https://sight.youdao.com/>

Ver también:

Traducción automática por API de NetEase Sight, Traducción automática, Sugerencias automáticas

2.16.82 MT_NETEASE_SECRET

App secret for the NetEase Sight API, you can register at <https://sight.youdao.com/>

Ver también:

Traducción automática por API de NetEase Sight, Traducción automática, Sugerencias automáticas

2.16.83 MT_TMSERVER

URL where tmserver is running.

Ver también:

tmserver, Traducción automática, Sugerencias automáticas, tmserver

2.16.84 MT_YANDEX_KEY

API key for the Yandex Translate API, you can register at <https://yandex.com/dev/translate/>

Ver también:

Yandex Translate, Traducción automática, Sugerencias automáticas

2.16.85 MT_YOUDAO_ID

Client ID for the Youdao Zhiyun API, you can register at <https://ai.youdao.com/product-fanyi-text.s>.

Ver también:

Youdao Zhiyun API machine translation, Traducción automática, Sugerencias automáticas

2.16.86 MT_YOUDAO_SECRET

Client secret for the Youdao Zhiyun API, you can register at <https://ai.youdao.com/product-fanyi-text.s>.

Ver también:

Youdao Zhiyun API machine translation, Traducción automática, Sugerencias automáticas

2.16.87 MT_SAP_BASE_URL

API URL to the SAP Translation Hub service.

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.88 MT_SAP_SANDBOX_APIKEY

API key for sandbox API usage

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.89 MT_SAP_USERNAME

Your SAP username

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.90 MT_SAP_PASSWORD

Your SAP password

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.91 MT_SAP_USE_MT

Whether to also use machine translation services, in addition to the term database. Possible values: True or False

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.92 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*.

2.16.93 PAGURE_CREDENTIALS

Nuevo en la versión 4.3.2.

List for credentials for Pagure servers.

Consejo: Use this in case you want Weblate to interact with more of them, for single Pagure endpoint stick with `PAGURE_USERNAME` and `PAGURE_TOKEN`.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.94 PAGURE_USERNAME

Nuevo en la versión 4.3.2.

Pagure username used to send merge requests for translation updates.

Ver también:

PAGURE_CREDENTIALS, *Pagure*

2.16.95 PAGURE_TOKEN

Nuevo en la versión 4.3.2.

Pagure personal access token used to make API calls for translation updates.

Ver también:

PAGURE_CREDENTIALS, *Pagure*, *Pagure API*

2.16.96 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*

2.16.97 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*

2.16.98 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`

2.16.99 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` enabled 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 URL for authentication.

Ver también:

`REGISTRATION_OPEN`, *Autenticación*

2.16.100 REGISTRATION_CAPTCHA

A value of either `True` or `False` indicating whether registration of new accounts is protected by CAPTCHA. This setting is optional, and a default of `True` will be assumed if it is not supplied.

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.

2.16.101 REGISTRATION_EMAIL_MATCH

Nuevo en la versión 2.17.

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$"
```

2.16.102 REGISTRATION_OPEN

Whether registration of new accounts is currently permitted. This optional setting can remain the default `True`, or changed to `False`.

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*

2.16.103 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

2.16.104 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](#).

2.16.105 SENTRY_DSN

Nuevo en la versión 3.9.

Sentry DSN to use for *Collecting error reports*.

Ver también:

Django integration for Sentry

2.16.106 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](#)

2.16.107 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.

2.16.108 SITE_DOMAIN

Configura el dominio del sitio. Es necesario para producir los URL absolutos correctos en distintos ámbitos (por ejemplo, en los mensajes de activación, las notificaciones o los suministros RSS).

Si Weblate se está ejecutando en algún puerto no estándar, inclúyalo aquí también.

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: Esta configuración debe contener solo el nombre de dominio. Para configurar el protocolo (activar y exigir HTTPS, p. ej.), utilice `ENABLE_HTTPS`, y para cambiar el URL, utilice `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`

2.16.109 SITE_TITLE

El título que se utilizará en el sitio web y los mensajes de correo que se envíen.

2.16.110 SPECIAL_CHARS

Caracteres adicionales que incluir en el teclado visual; *Visual keyboard*.

El valor predeterminado es:

```
SPECIAL_CHARS = ("\t", "\n", "...")
```

2.16.111 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` and in this case it only works in case 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"
```

2.16.112 STATUS_URL

The URL where your Weblate instance reports its status.

2.16.113 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.

2.16.114 UPDATE_LANGUAGES

Nuevo en la versión 4.3.2.

Controls whether languages database should be updated when running database migration and is enabled by default. This setting has no effect on invocation of `setuplang`.

Ver también:

Built-in language definitions

2.16.115 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.

2.16.116 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

2.16.117 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
```

2.16.118 WEBLATE_ADDONS

List of addons available for use. To use them, they have to be enabled for a given translation component. By default this includes all built-in addons, when extending the list you will probably want to keep existing ones enabled, for example:

```
WEBLATE_ADDONS = (
    # Built-in addons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.LanguangeConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.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",
    # Addon you want to include
    "weblate.addons.example.ExampleAddon",
)
```

Nota: Removing the addon from the list does not uninstall it from the components. Weblate will crash in that case. Please uninstall addon from all components prior to removing it from this list.

Ver también:

Complementos, `DEFAULT_ADDONS`

2.16.119 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

2.16.120 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

2.16.121 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

2.16.122 WEBSITE_REQUIRED

Defines whether *Sitio web del proyecto* has to be specified when creating a project. Turned 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 © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://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/>.
#

import os
import platform
from logging.handlers import SysLogHandler

#
# 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,
    },
}

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Data directory
DATA_DIR = os.path.join(BASE_DIR, "data")

# 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", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Tagbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),

```

(continué en la próxima página)

(proviene de la página anterior)

```

    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("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 format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(DATA_DIR, "static")

# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.

```

(continué en la próxima página)

(proviene de la página anterior)

```

# You can generate it using weblate/examples/generate-secret-key
SECRET_KEY = ""

_TEMPLATE_LOADERS = [
    "django.template.loaders.filesystem.Loader",
    "django.template.loaders.app_directories.Loader",
]
if not DEBUG:
    _TEMPLATE_LOADERS = [("django.template.loaders.cached.Loader", _TEMPLATE_
↪LOADERS)]
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",
            ],
            "loaders": _TEMPLATE_LOADERS,
        },
    },
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth",
    # "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_BITBUCKET_KEY = ""

```

(continué en la próxima página)

(proviene de la página anterior)

```

SOCIAL_AUTH_BITBUCKET_SECRET = ""
SOCIAL_AUTH_BITBUCKET_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
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",)

```

(continué en la próxima página)

(proviene de la página anterior)

```

SOCIAL_AUTH_SLUGIFY_USERNAMES = True
SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

# Password validation configuration
AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator" # noqa: E501, pylint: disable=line-too-long
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {"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",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [

```

(continué en la próxima página)

(proviene de la página anterior)

```

# 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",
"compressor",
"rest_framework",
"rest_framework.authtoken",
"django_filters",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↳WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - you can also choose "logfile" to log into separate file
#   after configuring it below

# Detect if we can connect to syslog
HAVE_SYSLOG = False
if platform.system() != "Windows":
    try:
        handler = SysLogHandler(address="/dev/log", facility=SysLogHandler.LOG_
↳LOCAL2)
        handler.close()
        HAVE_SYSLOG = True
    except OSError:
        HAVE_SYSLOG = False

```

(continué en la próxima página)

```

if DEBUG or not HAVE_SYSLOG:
    DEFAULT_LOG = "console"
else:
    DEFAULT_LOG = "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",
        },
        "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",
    # },
    },

```

(proviene de la página anterior)

```

"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",
    # },
    "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.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",
    # "weblate.machinery.netease.NeteaseSightTranslation",
    # "weblate.machinery.tmserver.AmagamaTranslation",
    # "weblate.machinery.tmserver.TMServerTranslation",
    # "weblate.machinery.yandex.YandexTranslation",
    # "weblate.machinery.saptranslationhub.SAPTranslationHub",
    # "weblate.machinery.youdao.YoudaoTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APY server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

```

(continué en la próxima página)

(proviene de la página anterior)

```

# 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
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

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

# 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

```

(continué en la próxima página)

(proviene de la página anterior)

```

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
# Sent referrrrer 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

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

```

(continué en la próxima página)

(proviene de la página anterior)

```

# 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_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PerlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.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.qt.QtFormatCheck",
#     "weblate.checks.qt.QtPluralCheck",
#     "weblate.checks.ruby.RubyFormatCheck",
#     "weblate.checks.consistency.PluralsCheck",
#     "weblate.checks.consistency.SamePluralsCheck",
#     "weblate.checks.consistency.ConsistencyCheck",
#     "weblate.checks.consistency.TranslatedCheck",
#     "weblate.checks.chars.EscapedNewlineCountingCheck",

```

(continué en la próxima página)

(proviene de la página anterior)

```

# "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",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LanguaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",
#     "weblate.addons.generate.GenerateFileAddon",
#     "weblate.addons.generate.PseudolocaleAddon",
#     "weblate.addons.json.JSONCustomizeAddon",
#     "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"

```

(continué en la próxima página)

```

# 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",
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
}

# Store sessions in cache
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# Store messages in session
MESSAGE_STORAGE = "django.contrib.messages.storage.session.SessionStorage"

# REST framework settings for API
REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth` permissions,
    # or allow read-only access for unauthenticated users.
    "DEFAULT_PERMISSION_CLASSES": [
        # Require authentication for login required sites
        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "rest_framework.authentication.SessionAuthentication",
    ),
    "DEFAULT_THROTTLE_CLASSES": (
        "weblate.api.throttling.UserRateThrottle",
        "weblate.api.throttling.AnonRateThrottle",
    ),
    "DEFAULT_THROTTLE_RATES": {"anon": "100/day", "user": "5000/hour"},
    "DEFAULT_PAGINATION_CLASS": ("rest_framework.pagination.PageNumberPagination"),
    "PAGE_SIZE": 20,
    "VIEW_DESCRIPTION_FUNCTION": "weblate.api.views.get_view_description",
}

```

(continué en la próxima página)

(proviene de la página anterior)

```

    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL
FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = [
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": ↪
    ↪ True},
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": ↪
    ↪ False},
]

# Require login for all URLs
if REQUIRE_LOGIN:
    LOGIN_REQUIRED_URLS = (r"/(.*)$",)

# In such case you will want to include some of the exceptions
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
#     rf"{URL_PREFIX}/accounts/(.*)$", # Required for login
#     rf"{URL_PREFIX}/admin/login/(.*)$", # Required for admin login
#     rf"{URL_PREFIX}/static/(.*)$", # Required for development mode
#     rf"{URL_PREFIX}/widgets/(.*)$", # Allowing public access to widgets
#     rf"{URL_PREFIX}/data/(.*)$", # Allowing public access to data exports
#     rf"{URL_PREFIX}/hooks/(.*)$", # Allowing public access to notification hooks
#     rf"{URL_PREFIX}/healthz/$", # Allowing public access to health check
#     rf"{URL_PREFIX}/api/(.*)$", # Allowing access to API
#     rf"{URL_PREFIX}/js/i18n/$", # JavaScript localization
#     rf"{URL_PREFIX}/contact/$", # Optional for contact form
#     rf"{URL_PREFIX}/legal/(.*)$", # Optional for legal app
# )

# 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_SCHEDULE_FILENAME = os.path.join(DATA_DIR, "celery", "beat-schedule")
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"},
}

```

(continué en la próxima página)

(proviene de la página anterior)

```
"weblate.memory.tasks.*": {"queue": "memory"},
}

# 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 = None
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 or pip3 installer, or by using the `./setup.py` script, the **weblate** is installed to your path (or virtualenv path), from where you can use it to control Weblate:

```
weblate list_versions
```

For the Docker image, the script is installed like above, and you can run it using **docker exec**:

```
docker exec --user weblate <container> weblate list_versions
```

For **docker-compose** the process is similar, you just have to use **docker-compose exec**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

In case you need to pass it a file, you can temporary add a volume:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/
↪users.json
```

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>

Nuevo en la versión 2.5.

Imports a translation from the file to use as a suggestion for the given translation. It skips duplicated translations; only different ones are added.

--author USER@EXAMPLE.COM

E-mail of author for the suggestions. This user has to exist prior to importing (you can create one in the admin interface if needed).

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>

Nuevo en la versión 2.5.

Distinto en la versión 4.6: Added parameter for translation mode.

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.

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 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).

Distinto en la versión 2.9: Added parameters `--username`, `--email`, `--name` and `--update`.

2.18.10 dump_memory

weblate dump_memory

Nuevo en la versión 2.20.

Export a JSON file containing Weblate Translation Memory content.

Ver también:

Memoria de traducción, Weblate Translation Memory Schema

2.18.11 dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by *importuserdata*

Consejo: Esto resulta útil cuando hay que migrar o fusionar instalaciones de Weblate.

2.18.12 import_demo

weblate import_demo

Nuevo en la versión 4.1.

Creates a demo project with components based on <<https://github.com/WeblateOrg/demo>>.

Puede ser de utilidad si está desarrollando Weblate.

2.18.13 import_json

weblate import_json <json-file>

Nuevo en la versión 2.7.

Batch import of components based on JSON data.

The imported JSON file structure pretty much corresponds to the component object (see `GET /api/components/(string:project)/(string:component)/`). You have to include the name and filemask fields.

--project PROJECT

Specifies where the components will be imported from.

--main-component COMPONENT

Use the given VCS repository from this component for all of them.

--ignore

Skip (already) imported components.

--update

Update (already) imported components.

Distinto en la versión 2.9: The parameters `--ignore` and `--update` are there to deal with 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:

`import_memory`

2.18.14 import_memory

weblate import_memory <file>

Nuevo en la versión 2.20.

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

2.18.15 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* addon, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on filemask.

<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 [import_json](#).

2.18.16 importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by [dumpuserdata](#)

2.18.17 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
```

2.18.18 install_addon

Nuevo en la versión 3.2.

weblate install_addon --addon ADDON <project|project/component>

Installs an addon to a set of components.

--addon ADDON

Name of the addon to install. For example `weblate.gettext.customize`.

--configuration CONFIG

JSON encoded configuration of an addon.

--update

Update the existing addon configuration.

You can either define which project or component to install the addon 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 --config '{"width": -1}' --
↪update --all
```

Ver también:

[Complementos](#)

2.18.19 list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names.

This is used to generate <https://wiki.110n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>.

2.18.20 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.

2.18.21 list_versions

weblate list_versions

Lists all Weblate dependencies and their versions.

2.18.22 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.

2.18.23 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:

`unlock_translation`

2.18.24 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.

2.18.25 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.

2.18.26 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:

[*lock_translation*](#)

2.18.27 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*](#)

2.18.28 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).

2.18.29 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.

2.18.30 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 polling by `updategit`.

2.19 Anuncios

Distinto en la versión 4.0: En versiones anteriores, esta prestación se denominaba «mensajes en pizarra».

Provide info to your translators by posting announcements, site-wide, per project, component, or language.

Announce the purpose, deadlines, status, or specify targets for translation.

Los usuarios recibirán una notificación por cada anuncio en los proyectos que monitorizan (salvo que decidan no recibirlas).

This can be useful for various things from announcing the purpose of the website to specifying targets for translations.

The announcements can be posted on each level in the *Manage* menu, using *Post announcement*:

Translations will be used only if they reach 60%.

Components Languages Info Search Insights Files Tools **Manage** Share Not watching

Post announcement

Message

You can use Markdown and mention users by @username.

Category

Info (light blue)

Category defines color used for the message.

Expiry date

mm/dd/yyyy

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ **Notify users**

The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.

Add

Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate

It can be also added using the admin interface:

Weblate administration

WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

[Home](#) · [Weblate translations](#) · [Announcements](#) · [Add Announcement](#)

Add Announcement

Required fields are marked in bold.

Message:

Translations will be used only if they reach 60%.

You can use Markdown and mention users by @username.

Project: WeblateOrg  

Component: -----  

Language: -----  

Category: Info (light blue) 
Category defines color used for the message.

Expiry date: Today 
The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ **Notify users**

Save and add another Save and continue editing SAVE

The announcements are then shown based on their specified context:

Ningún contexto especificado

Shown on dashboard (landing page).

Proyecto especificado

Shown within the project, including all its components and translations.

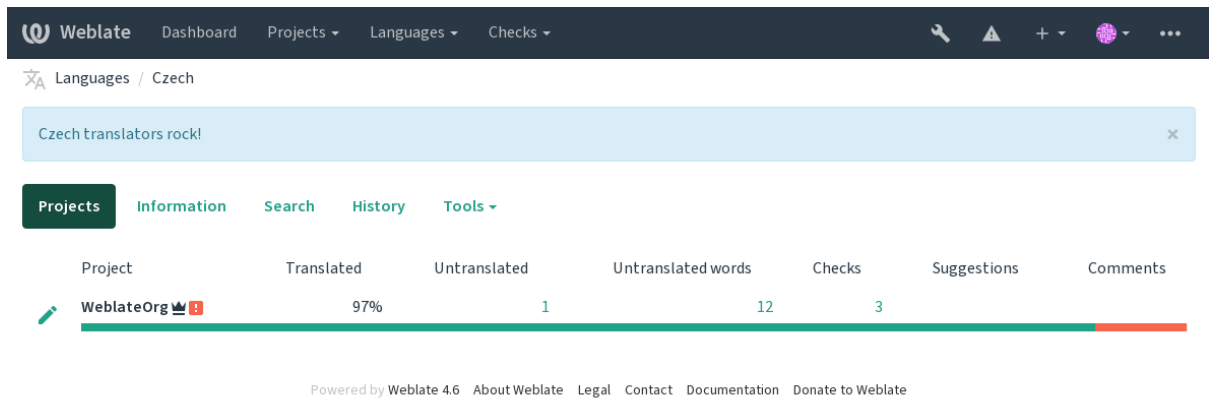
Componente especificado

Shown for a given component and all its translations.

Idioma especificado

Shown on the language overview and all translations in that language.

This is how it looks on the language overview page:



2.20 Listas de componentes

Specify multiple lists of components to appear as options on the user dashboard, from which users can pick one as their default view. See *Cuadro de mando* to learn more.

Distinto en la versión 2.20: A status will be presented for each component list presented on the dashboard.

The names and content of component lists can be specified in the admin interface, in *Component lists* section. Each component list must have a name that is displayed to the user, and a slug representing it in the URL.

Distinto en la versión 2.13: Change dashboard settings for anonymous users from the admin interface, altering what dashboard is presented to unauthenticated users.

2.20.1 Listas de componentes automáticas

Nuevo en la versión 2.13.

Add components to the list automatically based on their slug by creating *Automatic component list assignment* rules.

- Useful for maintaining component lists for large installations, or in case you want to have one component list with all components on your Weblate installation.

Consejo: Make a component list containing all the components of your Weblate installation.

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:

Weblate administration

WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Component lists · Add Component list

Add Component list

Required fields are marked in bold.

Component list name:

All components

Display name

URL slug:

all-components

Name used in URLs and filenames.

☒ Show on dashboard

When enabled this component list will be shown as a tab on the dashboard

Components:

Available components ⓘ

Filter

WeblateOrg/Django

WeblateOrg/Language names

WeblateOrg/WebblateOrg

Choose all ⓘ

Chosen components ⓘ

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE?
^.*\$	^.*\$	✕

+ Add another Automatic component list assignment

Save and add another

Save and continue editing

SAVE

2.21 Módulos opcionales de Weblate

Hay varios módulos opcionales disponibles para que los configure.

346

Capítulo 2. Documentación para administradores

2.21.1 Git exporter

Nuevo en la versión 2.10.

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

Nuevo en la versión 2.4.

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

Nuevo en la versión 2.15.

This is used on [Hosted Weblate](#) to provide required legal documents. It comes provided with blank documents, and you are expected to fill out the following templates in the documents:

legal/documents/tos.html Terms of service document

legal/documents/privacy.html Privacy policy document

legal/documents/summary.html Short overview of the terms of service and privacy policy

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:

- Suggestions from unauthenticated users
- Project and component descriptions and links

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:


[Weblate](#)
[Dashboard](#)
[Projects ▾](#)
[Languages ▾](#)
[Checks ▾](#)
[Register](#)
[Sign in](#)
...

[About Weblate](#) / [Weblate keys](#)

[About Weblate](#)
[Statistics](#)
[Keys](#)

SSH key

SSH key not available.

Commit signing

All commits made with Weblate are signed with the GPG key D3386879A59B6F9BD780039D5E3D09F5D28D1902, for which the corresponding public key is found below.

```

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGB8e4sBDADi7OXDBHNKpYGL0qwUwY25HkntNcbWhtocalTddUiGuafo/cG7
nJMwhfir98OMcfaXsc/tYmjii/phWtsK0cR4rYuZcQFERYyTcMRFAMZcUfqz8xx
DyNO05sZDAZtW/5Ra6n+Dg+FWJ0xPWZAsIjFs+V2NOIKdIAW05W4Tb4gp72qPQTE
Ts3iXoFb8uQNuyD6sxQegpzpJGlpe/nw/GzmqkvKAkJxx7XlyjgWaZgpANhbTXK
htKV/TyCP6XZbvRlvLaKWppNvMgFHC8aWjZyOmr/b3al42deewxJNa0DJyDMsx
hCrC2lAs4ysk/hZFo8lQ6+rabwXjw9CRJAGm/YGFwsovv4godCT16icjYC7sJaC5
z0dL9chSmMJMsZRB+5nZTdlvfqzeN342LkojAlz2VEJnmg8EHAU65kCfHl3v7MI
SWuPgJTcnEa3gUHFILWHn9ohl3jmUP2b57ZQzAlmISZy+jsm9UbqootQit6ecqlg
5rk6ljjvLeX+NKcAEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWIQTTOGH5pZtvm9eAA51ePQn10o0ZAgUCYHx7iwlBawULCQgHAgYV
CgkICwIEFgIDAQIeAQIXgAAKCRBePQn10o0ZAnxkC/sGm1V2CwpVwOI73lK8nli
JEhVHFrDeOwkXhN96hczUJqocZAmseUDgQ4eeY4YW55/QCGIH00wz3XoslCp00k
c6qFluBIUWjXUyTs9vTcf3eU5tcPxxaaXIVozkdAx1sZlvB7H7ODvuUYIQDJXJNk
6iPh7z6VW4R6+eLNG5z+nOii3h4l7l7PY//EgzQ4Sx7kdxAkKvT7S/nT7vx9IGHV

```

Powered by Weblate 4.6
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking `gpg`.

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	5	300	600
Password authentication on sign in	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

If a user fails to log in `AUTH_LOCK_ATTEMPTS` times, password authentication will be turned off on the account until having gone through the process of having its password reset.

The settings can be also applied in the Docker container by adding `WEBLATE_` prefix to the setting name, for example `RATELIMIT_ATTEMPTS` becomes `WEBLATE_RATELIMIT_ATTEMPTS`.

The API has separate rate limiting settings, see [API rate limiting](#).

Ver también:

[Rate limiting](#), [Running behind reverse proxy](#), [API rate limiting](#)

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 some custom Python code (called a module), a place to store it is needed, either in the system path (usually something like `/usr/lib/python3.7/site-packages/`) or in the Weblate directory, which is also added to the interpreter search path.

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`).

La estructura de su módulo debería lucir similar a esta:

```
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:

Managing static files (e.g. images, JavaScript, CSS), *Serving static files*

2.22.3 Comprobaciones de calidad, complementos y correcciones automáticas personalizadas

To install your code for *Correcciones automáticas personalizadas*, *Writing own checks* or *Escribir un complemento* 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",)

# Addons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)

```

Ver también:

Correcciones automáticas personalizadas, *Writing own checks*, *Escribir un complemento*, *Ejecutar secuencias de órdenes de un complemento*

2.23 Interfaz de gestión

The management interface offer administration settings under the `/manage/` URL. It is available for users signed in with admin privileges, accessible by using the wrench icon top right:

Weblate
Dashboard
Projects
Languages
Checks

Manage

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

Tools
Billing

Weblate support status

Weblate version
4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c

Support status
Community support

Purchase support package

Donate to Weblate

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Powered by Weblate 4.6
About Weblate
Legal
Contact
Documentation
Donate to Weblate

It includes basic overview of your 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*
- Alerts overview for all components, see alerts

2.23.1 La interfaz administrativa de Django

Advertencia: Will be removed in the future, as its use is discouraged—most features can be managed directly in 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 logs	+ Add	Change
Profiles	+ Add	Change
Verified emails	+ Add	Change
AUTH TOKEN		
Tokens	+ Add	Change
AUTHENTICATION		
Groups	+ Add	Change
Roles	+ Add	Change
Users	+ Add	Change
BILLING		
Billings	+ Add	Change
Invoices	+ Add	Change
Plans	+ Add	Change
FONTS		
Font groups	+ Add	Change
Fonts	+ Add	Change
LEGAL		
Agreements	+ Add	Change
PYTHON SOCIAL AUTH		
Associations	+ Add	Change
Nonces	+ Add	Change
User social auths	+ Add	Change
SCREENSHOTS		
Screenshots	+ Add	Change
TRANSLATION MEMORY		
Memorys	+ Add	Change
WEBLATE CONFIGURATION		
Settings	+ Add	Change
WEBLATE LANGUAGES		
Languages	+ Add	Change
WEBLATE TRANSLATIONS		
Announcements	+ Add	Change
Component lists	+ Add	Change
Components	+ Add	Change
Contributor agreements	+ Add	Change
Projects	+ Add	Change

Recent actions

My actions

None available

In the *Reports* section, you can check the status of your site, tweak it for *Puesta en marcha de entorno de producción*, or manage SSH keys used to access *Accessing repositories*.

Manage database objects under any of the sections. The most interesting one is probably *Weblate translations*, where you can manage translatable projects, see *Configuración de proyectos* and *Configuración de componentes*.

Weblate languages holds language definitions, explained further in *Language definitions*.

Añadir un proyecto

Adding a project serves as container for all components. Usually you create one project for one piece of software, or book (See *Configuración de proyectos* for info on individual parameters):

Weblate administration

WELCOME, **WEBLATE TEST** · [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Projects · Add Project

Add Project

Required fields are marked in bold.

Project name:

Display name

URL slug:

Name used in URLs and filenames.

Project website:

Main website of translated project.

Translation instructions:

You can use Markdown and mention users by @username.

☒ Set "Language-Team" header

Lets Weblate update the "Language-Team" file header of your project.

☒ Use shared translation memory

Uses the pool of shared translations between projects.

☒ Contribute to shared translation memory

Contributes to the pool of shared translations between projects.

Access control:

Protected

How to restrict access to this project is detailed in the documentation.

☐ Enable reviews

Requires dedicated reviewers to approve translations.

☐ Enable source reviews

Requires dedicated reviewers to approve source strings.

☒ Enable hooks

Whether to allow updating this repository by remote hooks.

Language aliases:

Comma-separated list of language code mappings, for example: en_GB:en,en_US:en

Save and add another

Save and continue editing

SAVE

Ver también:

Configuración de proyectos

Componentes bilingües

Once you have added a project, translation components can be added to it. (See [Configuración de componentes](#) for info regarding individual parameters):

Add Component

Open · Editable translations · Compare · Add component · Add component

Create new component

Add Component

Required fields are marked in bold.

Component name

 Language names

URL slug

 language-names
Name used in URLs and filenames

Project

 WhedonCyg +

Version control system

 Git

Select version control system 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/WhedonCyg/language-
URL of a repository; you will receive a language-component to share it with other components

Repository push URL:

URL of a push repository; pushing is turned off if empty

Repository browser:

 https://github.com/WhedonCyg/language-MachineTranslations/browseURL
Link to repository browser; use Github for source, Bitbucket and Cloud for Reviews and live placeholders

Expected repository URL:

URL of repository where users can fetch changes from Website

Source string bug-reporting address:

Email address for reports or errors in source strings. Leave empty for e-mails

Repository branch:

Repository branch to translate

Push branch:

Branch for pushing changes; leave empty to use repository branch

Flavors:

 website:language-to-wh7-t-L-C.MESSAGES-flv
Path of file to translate relative to repository root; use "/" instead of language code; for example "path/to/flavor/L-C_MESSAGES" or path

Monolingual base language file:

Filename of translation base file, containing all strings and their sources. It is recommended for monolingual translation formats.

☒ Edit base file

 Other users will be able to edit the base file for monolingual translations.

Intermediate language files:

Filename of intermediate translated or fix - to most cases this is a translation file provided by developers and/or used when creating actual source strings

Template for new translations:

 website:language-to-wh7-gdrgo.gpt
Filename of the used for creating new translations. For general choice .pot file

File format:

 gettext PC file

☐ Locked

 Subcomponent will not get any translated updates.

☒ Allow translation propagation

 Whether translation updates in other components will cause automatic translation in this one

☒ Run on suggestions

 Whether to allow translation suggestions at all.

☐ Suggestion voting

 Users can only vote for suggestions and can't make direct translations.

Advanced suggestions:

 0
Automatically accept suggestions with this number of votes; use 0 to turn it off

Translation flags:

Additional comma-separated flags to influence quality checks. Possible values can be found in the documentation.

Enforced checks:

List of checks which cannot 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.

Add 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 Website.

☐ Merge strings

 Enable adding and merging existing strings from Website. If your strings are extracted from the source code or managed externally, you probably want to keep it disabled.

Merge style:

 Rebase
Define whether Website should merge the upstream repository or rebased entirely.

Current message when translating:

Translated using Website {0} (language_name {0})
Currently translated at {0} state translated, percent {0}% ({0} state translated {0} of {0} state at {0} strings)
Translation {project_name {0}} (component_name {0})
Translate URL: {url{0}}

You can use template language for website info; please consult the documentation for more details.

Current message when adding translation:

Added translation using Website {0} (language_name {0})

You can use template language for website info; please consult the documentation for more details.

Current message when removing translation:

Deleted translation using Website {0} (language_name {0})

You can use template language for website info; please consult the documentation for more details.

Current message when merging translation:

Merged branch {0} (component_branch_branch {0}) into Website.

You can use template language for website info; please consult the documentation for more details.

Current message when editor makes a change:

Update translation file
Updating {translation_name {0}} back to Website
Translation {project_name {0}} (component_name {0})
Translate URL: {url{0}}

You can use template language for website info; please consult the documentation for more details.

☒ Push on commit

 Whether the repository should be pushed upon 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:

 %wh%flv%
Regular expression used to filter translation files when searching for flavors.

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.

There are projects:

 WhedonCyg +

Choose additional projects where this component will be tested. Add new "Control" or "Component" as alias, to inherit more than one.

☐ Use as a glossary

Obseary color:

 Silver

Save and calculate

Save and continue editing

Save

Ver también:

Configuración de componentes, Formatos bilingües y monolingües

Componentes monolingües

For easier translation of these, provide a template file containing the mapping of message IDs to its respective source language (usually English). (See *Configuración de componentes* for info regarding individual parameters):

Webiate administration

WELCOME WEBLATE TEAM | RETURN TO WEBLATE | DOCUMENTATION | CHANGE PASSWORD | LOG OUT

Home | WEBLATE TRANSLATIONS | Components | ADD COMPONENT

ADD NEW COMPONENT

Required fields are marked with *

Component name:

Android

URL slug:

android

Project:

WebIdeology

Version control system:

Git

Source code repository:

weblate://weblate.org/language-names

Repository push URL:

Repository browser:

Expected repository URL:

Source string logging reporting address:

Repository branch:

Push branch:

Flormask:

app/src/main/res/values-strings.xml

Monolingual base language file:

app/src/main/res/values-strings.xml

Use base file

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file:

Template for new translations:

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.

Autosuggest suggestions:

0

Translation flags:

Enforced checks:

Translation license:

MIT License

Contribution agreement:

Adding new translation:

Choose new language file

Language code style:

Default based on the file format

Whisper strings

Enables adding and removing string groups from Weblate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.

Merge style:

Rebase

Current message when handling:

Translated using Weblate (% language, name %)
Currently translated at (% state,translated,percent %%) (% state,translated % of % state,all % strings)
Translation (% project, name %%) component_name %)
Translate URL: % at %

Current message when adding translation:

Add/translation using Weblate (% language, name %)

Current message when removing translation:

Deleted translation using Weblate (% language, name %)

Current message when merging translation:

Merge branch (% component_name, branch %) into Weblate.

Current message when adding makes a change:

Update translation file
Updated by (% address, name %) track in Weblate.
Translation (% project, name %%) component_name %)
Translate URL: % at %

Push on commit

Whether new requests should be pushed (approved) or only comment.

Age of changes to commit:

24

Lock on error

Whether the component should be locked on repository errors.

Source language:

English

Language filter:

all

Variables regular expression:

Priority:

Medium

Restricted component

Restrict access to the component to only those explicitly given permission.

Share in projects:

WebIdeology

Use as a glossary

Glossary order:

Star

Click and add another

Save and continue editing

Save

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.

The screenshot shows the Weblate dashboard interface. At the top is a dark navigation bar with the Weblate logo and links for Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with a row of tabs: Weblate status (highlighted in green), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, and Appearance. Under the 'Weblate status' tab, there are sub-tabs for Tools and Billing. The main content area displays the 'Weblate support status' section, which includes the Weblate version (4.6) and a unique hash, the support status (Community support), and buttons for purchasing a support package or donating to Weblate. Below this is the 'Activate support package' section, which contains a text input field for an activation token and an 'Activate' button.

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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
- *Weblate provisioned backup storage*
- *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.

Nota: Esta función está disponible actualmente en fase beta temprana.

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.

Getting listed

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:

Manage

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

Tools Billing

Weblate support status ⓘ

Weblate version 4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c

Support status Community support

Discover Weblate Your Weblate is not listed on weblate.org [Browse discovery](#)

Enable discovery

Manage support package **Purchase support package** **Donate to Weblate**

Activate support package ⓘ

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate **Purchase support package**

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Listing your server without a support subscription in Discover Weblate:

1. Register yourself at <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:

Manage

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

Tools Billing

Weblate support status ⓘ

Weblate version 4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c

Support status Community support

Discover Weblate Your Weblate is not listed on weblate.org [Browse discovery](#)

[Enable discovery](#)

[Manage support package](#) [Purchase support package](#) [Donate to Weblate](#)

Activate support package ⓘ

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

[Activate](#) [Purchase support package](#)

Powered by Weblate 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Customizing listing

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 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.

2.25.2 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](#)

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*
- *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 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](#).

3.1.4 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.

Backers who have funded 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

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*.

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 .
```

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
./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 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:

```
./rundev.sh test --failfast weblate.trans
```

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
```

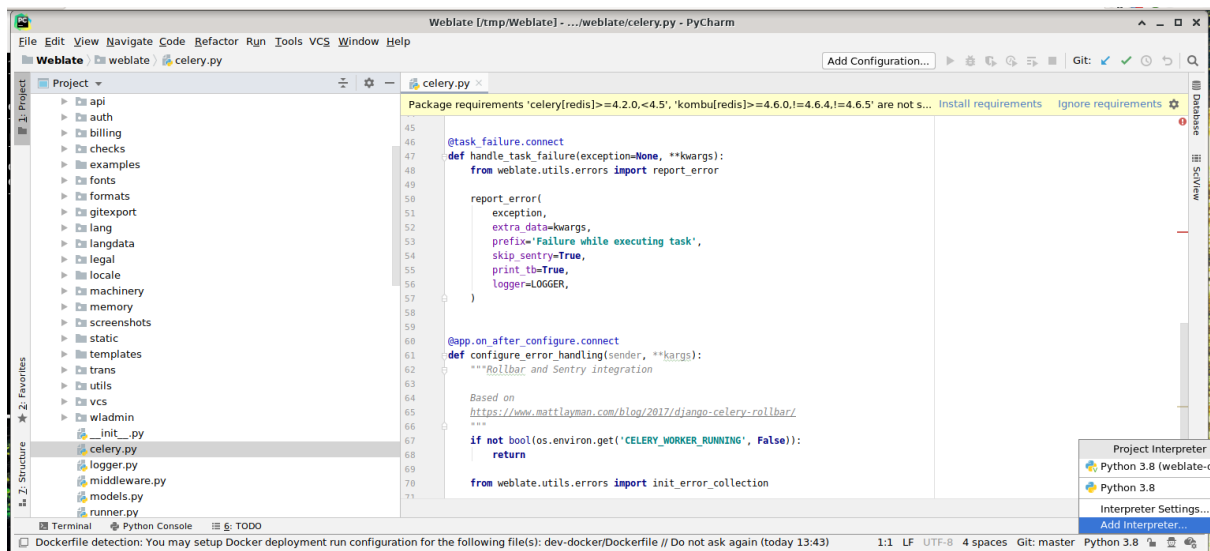
Ejecutar la secuencia de órdenes sin argumentos recreará el contenedor de Docker y lo reiniciará.

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.

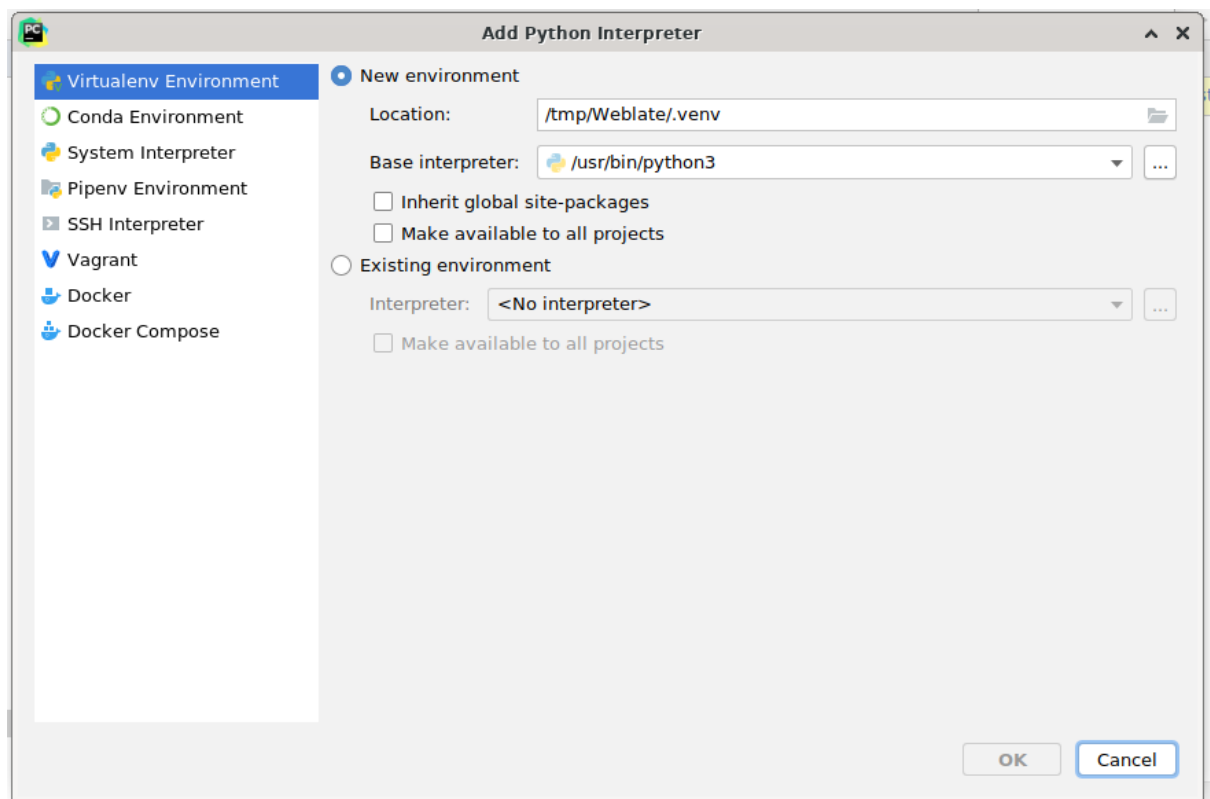
3.2.4 Trabajar en Weblate con PyCharm

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:

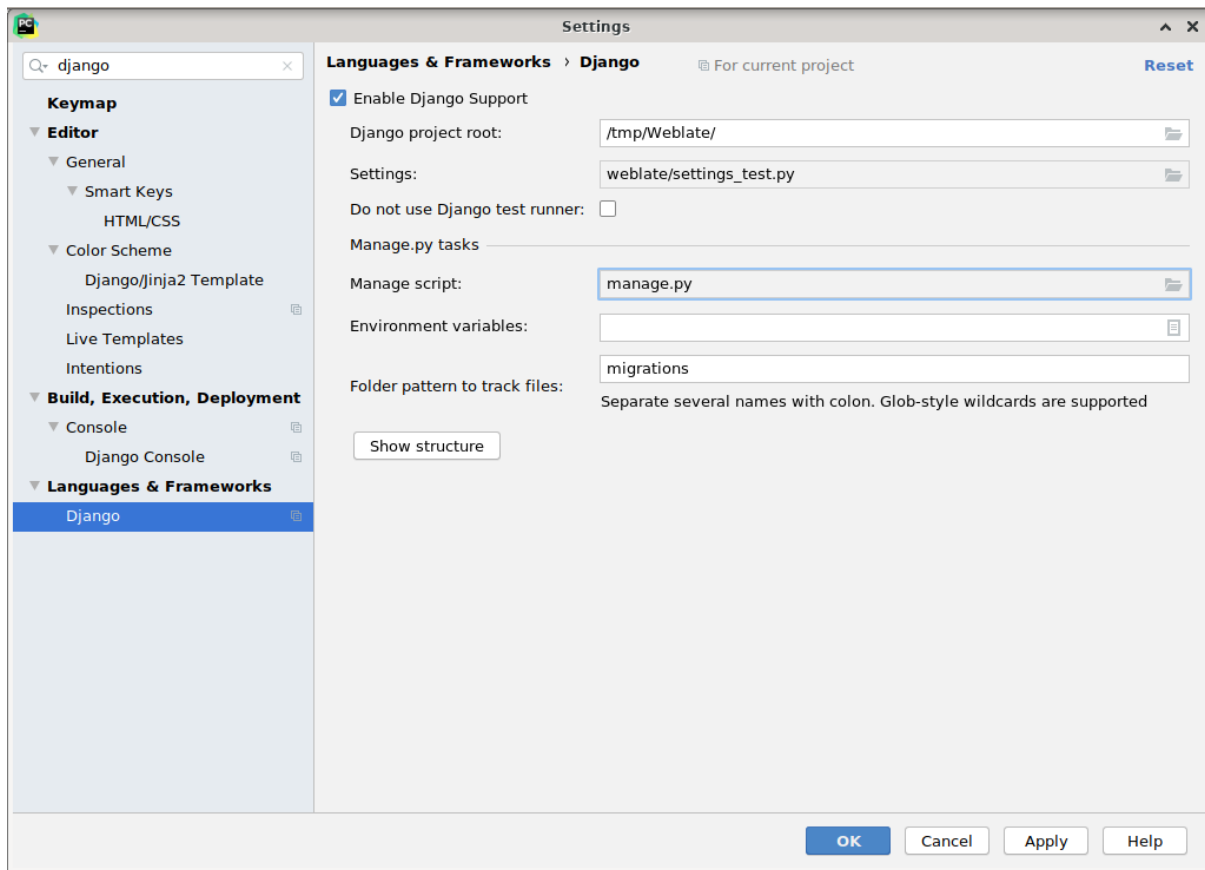


You can either choose to let PyCharm create the virtualenv for you, or select an already existing one:



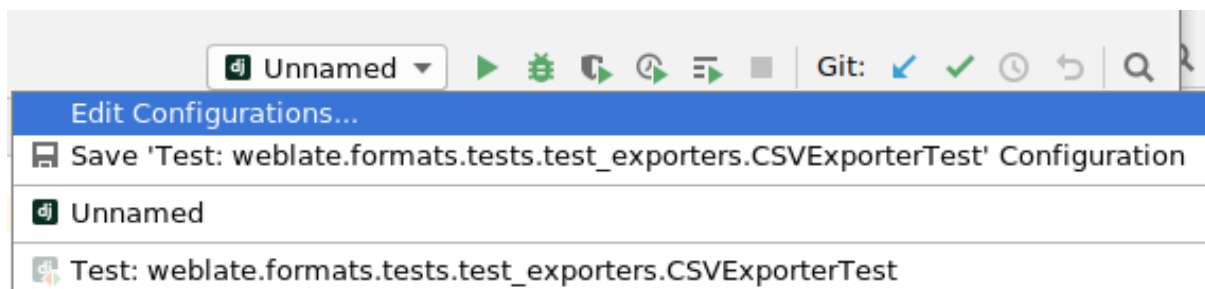
Don't forget to install the dependencies once the interpreter is set: Either through the console (the console from the IDE will directly use your virtualenv by default), or through the interface when you get a warning about missing dependencies.

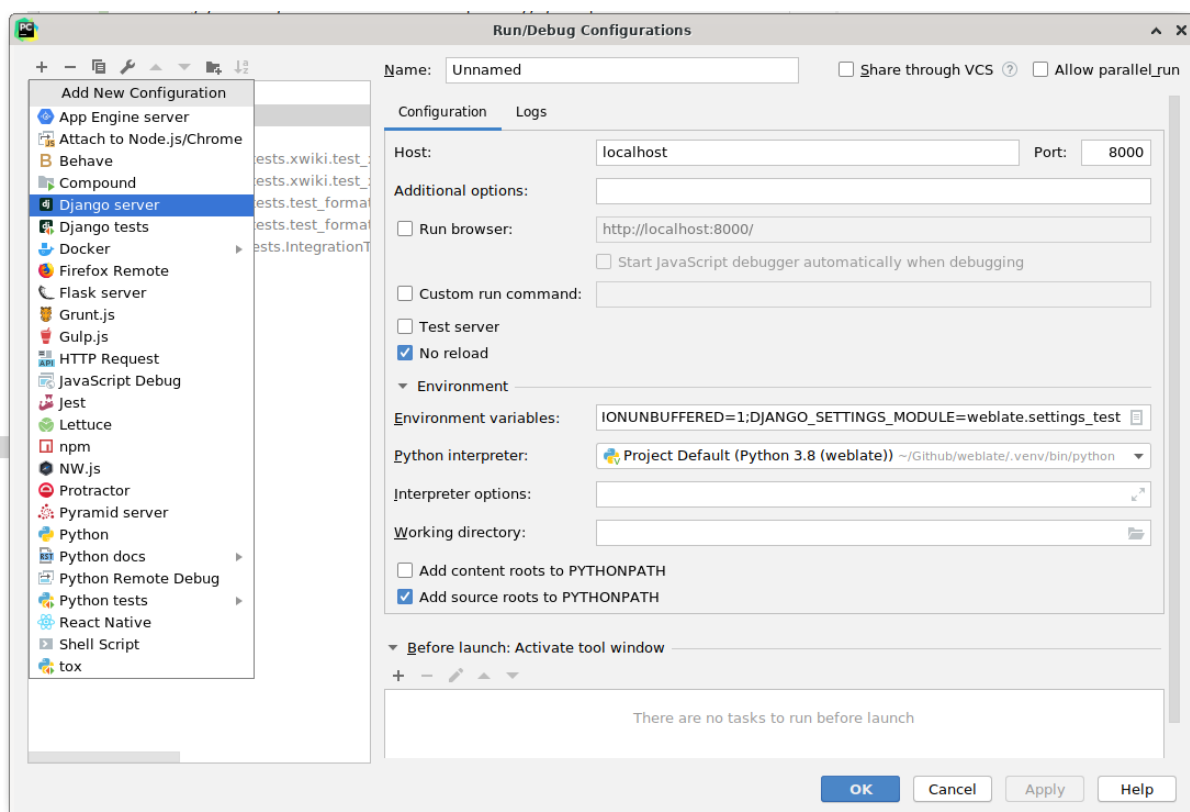
The second step is to set the right info to use Django natively inside PyCharm: The idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of the Django project and the path to its settings:



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.

3.2.5 Bootstrapping your devel instance

You might want to use `import_demo` to create demo translations and `createadmin` to make an admin user.

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 Principios de seguridad por naturaleza

Cualquier código escrito para Weblate debe crearse teniendo en mente los *principios de seguridad por naturaleza*.

3.3.2 Coding standard

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**. Weblate 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 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 browser. This is useful to debug issues in the web interface, but not suitable for production environment as it has performance consequences and might leak private data.

Ver también:

Desactivar el modo de depuración

3.4.2 Registros de Weblate

Weblate can produce detailed logs of what is going 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 own logs as well. The example system-wide setups log to several files under `/var/log/celery/`.

Docker containers log to their output (as usual in the Docker world), so you can look at the logs using `docker-compose logs`.

Ver también:

Configuración de muestra contains `LOGGING` configuration.

3.4.3 Not processing background tasks

Lot of things happen in background Celery workers. In case things like sending out e-mails or component removal does not work, there might be some issue with it.

Elementos que revisar en este caso:

- Check Celery process is running, see *Tareas en segundo plano con Celery*
- Check Celery queue status either in *Interfaz de gestión* or using `celery_queues`
- Look into 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 using *Interfaz de gestión* under the *Tools* tab.

These send e-mail 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. The easiest way to achieve this is 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 situation, 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.

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`

Addons 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`

Built-in translation memory, see [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 Desarrollo de complementos

Complementos are way to customize localization workflow in Weblate.

```
class weblate.addons.base.BaseAddon (storage=None)
```

```
classmethod can_install (component, user)
```

Check whether addon is compatible with given component.

```
configure (settings)
```

Save configuration.

```
daily (component)
```

Actuador desencadenado diariamente.

```
classmethod get_add_form (user, component, **kwargs)
```

Return configuration form for adding new addon.

```
get_settings_form (user, **kwargs)
```

Return configuration form for this addon.

```
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 addon 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)
```

Hook triggered before changes are committed to the repository.

```
pre_push (component)
```

Hook triggered before repository is pushed upstream.

```
pre_update (component)
```

Hook triggered before repository is updated from upstream.

```
save_state ()
```

Save addon state information.

stay_on_create = False

Clase de base para complementos de Weblate.

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.

He aquí un complemento de ejemplo:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://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/>.
#

from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT

class ExampleAddon(BaseAddon):
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {"file_format": {"po", "po-mono"}}
    # List of events addon should receive
    events = (EVENT_PRE_COMMIT,)
    # Addon unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example addon")
    # Detailed addon description
    description = _("This addon 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.

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 Regionalización

Should you need any user visible text in the frontend code, it should be localizable. In most cases all you need is to wrap your text inside `gettext` function, but there are more complex features available:

```
document.write(gettext('this is to be translated'));

var object_count = 1 // or 0, or 2, or 3, ...
s = gettext('literal for the singular case',
            'literal for the plural case', object_count);

fmts = gettext('There is %s object. Remaining: %s',
               'There are %s objects. Remaining: %s', 11);
s = interpolate(fmts, [11, 20]);
// s is 'There are 11 objects. Remaining: 20'
```

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

Weblate [issue tracker](#) is hosted at 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@cihar.com. 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`:

```
# Simple way to configure test database from environment

# 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
./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:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
```

(continué en la próxima página)

(proviene de la página anterior)

```
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	<i>The Translation Memory Item</i>
type	object
properties	
• category	<i>The String Category</i> 1 is global, 2 is shared, 10000000+ are project specific, 20000000+ are user specific type integer examples 1 minimum 0 default 1
• origin	<i>The String Origin</i> Filename or component name type string examples test.tmx proyecto/componente default
• source	<i>The Source String</i> type string examples Hello minLength 1 default
• source_language	<i>The Source Language</i> ISO 639-1 / ISO 639-2 / IETF BCP 47 type string

continué en la próxima página

Tabla 1 – proviene de la página anterior

		examples	es
		pauta	^[^]+\$
		default	
	• target	<i>The Target String</i>	
		type	<i>string</i>
		examples	Ahoj
		minLength	1
		default	
	• target_language	<i>The Target Language</i>	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		type	<i>string</i>
		examples	cs
		pauta	^[^]+\$
		default	
	additionalProperties		Falso
definiciones			

Ver también:

Memoria de traducción, dump_memory, import_memory

3.10.2 Exportación de datos de usuarios de Weblate

https://weblate.org/schemas/weblate-userdata.schema.json			
type	object		
properties			
• basic	<i>Basic</i>		
	type	object	
	properties		
	• username	<i>Nombre de usuario</i>	
		type	string
		examples	administrador
		default	
	• full_name	<i>Nombre completo</i>	
		type	string
		examples	Weblate Admin
		default	
	• email	<i>Correo electrónico</i>	
		type	string
		examples	noreply@example.com
		default	
	• date_joined	<i>Fecha de incorporación</i>	
		type	string
		examples	2019-11-18T18:53:54.862Z
		default	
• profile	<i>Perfil</i>		
	type	object	
	properties		
	• language	<i>Idioma</i>	
		type	string
		examples	cs
		pauta	^.*\$
		default	
	• suggested	<i>Number of suggested strings</i>	
		type	integer

continué en la próxima página

Tabla 2 – proviene de la página anterior

		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		
	• hi-de_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		
	• hi-de_source_secondary	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	^.*\$		
		default			
	• dashboard_view	Vista predeterminada del cuadro de mando			
		type	integer		
		examples	1		
		default	0		
	• dashboard_components_list	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		

continué en la próxima página

Tabla 2 – proviene de la página anterior

	• secondary_languages	Idiomas secundarios		
		type		array
		default		
		items	Código de idioma	
			type	string
			examples	sk
			pauta	^.*\$
			default	
	• monitorizado	Proyectos monitorizados		
		type		array
		default		
		items	«Slug» del proyecto	
			type	string
			examples	weblate
			pauta	^.*\$
			default	
• Registro de auditoría	Registro de auditoría			
	type		array	
	default			
	items	Elementos		
		type	object	
			properties	
		• Dirección	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	^.*\$
			default	
		• cronomarcador	Cronomarcador	
			type	string
			examples	2019-11-18T18:58:30.845Z
			pauta	^.*\$
			default	
		• actividad	Actividad	
			type	string
	examples		acceso	
	pauta		^.*\$	
	default			
definiciones				

Ver también:

Perfil de usuario, 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 -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`.

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 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.

3.12.2 Docker container security

Los contenedores para Docker se analizan con [Anchore](#) y [Trivy](#).

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 Acerca de Weblate

3.13.1 Objetivos del proyecto

Web-based continuous localization tool with tight *Integración de control de versiones* supporting a wide range of *Formatos de archivo admitidos*, making it easy for translators to contribute.

3.13.2 Nombre del proyecto

«Weblate» es una palabra híbrida creada a partir de las voces inglesas «web» y «translate».

3.13.3 Sitio web del proyecto

The landing page is <https://weblate.org> and a cloud hosted service at <https://hosted.weblate.org>. This documentation can be found on <https://docs.weblate.org>.

3.13.4 Logotipos del proyecto

The project logos and other graphics is available in <https://github.com/WeblateOrg/graphics/> repository.

3.13.5 Leadership

This project is maintained by Michal Čihař <michal@cihar.com>.

3.13.6 Autores

Weblate was started by Michal Čihař <michal@cihar.com>. Since its inception in 2012, thousands of people have contributed.

3.14 Licencia

Copyright (C) 2012 - 2021 Michal Čihař <michal@cihar.com>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

4.1 Weblate 4.6

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 direct 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 addon 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*.
- Simplified state indication while listing strings.
- Password hashing now uses Argon2 by default.
- Simplified progress bars indicating translation status.
- Renamed *Add missing languages* to clarify the purpose.
- Fixed saving string state to XLIFF.
- Added language-wide search.
- Initial support for *Scaling horizontally* the Docker deployment.

[All changes in detail.](#)

4.2 Weblate 4.5.3

Released on April 1st 2021.

- Fixed metrics collection.
- Fixed possible crash when adding strings.
- Improved search query examples.
- Fixed possible loss of newly added strings on replace upload.

4.3 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.
- Added integration with *Descubrir Weblate*.
- Fixed checks behavior with read-only strings.

[All changes in detail.](#)

4.4 Weblate 4.5.1

Publicada el 5 de marzo de 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 glossaries.
- 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.
- Improved interface for customizing translation-file downloads

[All changes in detail.](#)

4.5 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.
- Se sumó un complemento de generación de pseudorregiones.
- 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.

[All changes in detail.](#)

4.6 Weblate 4.4.2

Publicada el 14 de enero de 2021.

- Se arregló el daño en un archivo MO distribuido.

4.7 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.
- El comportamiento del complemento de limpieza es ahora el esperable al tratar archivos HTML, ODF, IDML y RC de Windows.
- 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 Pagine.
- 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 addon breaking adding new strings.
- Se añadió una API para complementos.

[All changes in detail.](#)

4.8 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.
- Se creó un complemento para quitar las cadenas vacías de los archivos de traducción.
- Ahora se utiliza el editor CodeMirror para las traducciones.
- Coloración sintáctica en el editor de traducciones para XML, HTML, Markdown y reStructuredText.
- Highlight placeables in translation editor.
- 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.

[All changes in detail.](#)

4.9 Weblate 4.3.2

Publicada el 4 de noviembre de 2020.

- Fixed crash on certain component filemasks.
- 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.

[All changes in detail.](#)

4.10 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.

[All changes in detail.](#)

4.11 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 sustitutorios 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.

[All changes in detail.](#)

4.12 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.

4.13 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 addon for some XLIFF files.
- Se permite configurar la CDN en la imagen de Docker.

4.14 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.
- El complemento de concentración de Git puede generar renglones finales `Co-authored-by:`.
- 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 addon 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.

4.15 Weblate 4.1.1

Publicada el 19 de junio de 2020.

- Fixed changing autofix or addons 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](#).

4.16 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 addons to enable.
- Se reparó el atajo de teclado en el editor para pasar por alto las comprobaciones.
- Improved machine translation of strings with placeholders.
- 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.

4.17 Weblate 4.0.4

Publicada el 7 de mayo de 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.

4.18 Weblate 4.0.3

Publicada el 2 de mayo de 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.

4.19 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.
- Fixed XLIFF export with placeholders.
- 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.

4.20 Weblate 4.0.1

Publicada el 16 de abril de 2020.

- Se arregló la instalación del paquete desde PyPI.

4.21 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.
- Added check and highlighter for percent placeholders.
- Se muestran las comprobaciones fallidas de las sugerencias.
- Record source string changes in history.
- Upgraded Microsoft Translator to version 3 API.
- Reimplemented translation memory backend.
- 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 addon 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.

4.22 Serie 3.x de Weblate

4.22.1 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.
- Se repararon los enlaces a la documentación de los complementos.
- 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.

4.22.2 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.

4.22.3 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.

4.22.4 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 filemask 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 addon.
- 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 addons now do full sync of translations in Weblate.

4.22.5 Weblate 3.10.3

Publicada el 18 de abril de 2020.

- Compatibilidad con translate-toolkit 2.5.0.

4.22.6 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.
- Complemento nuevo para configurar el formato de YAML.
- Fixed same plurals check to not fire on single plural form languages.
- Se arregló la búsqueda por expresiones regulares en algunos campos.

4.22.7 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.

4.22.8 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.

4.22.9 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.

4.22.10 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.
- Added new generic check for placeholders.
- Added support for WebExtension JSON placeholders.
- 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 addon to bootstrap translations.
- Extend automatic translation to add suggestions.
- Display addon 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 addon 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.

4.22.11 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 addons.
- 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.
- Es posible clonar los complementos en los componentes detectados.
- 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.

4.22.12 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.

4.22.13 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 addon.
- Delay opening SMTP connection when sending notifications.
- Improved error logging.
- Allow custom location in MO generating addon.
- Added addons 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.

4.22.14 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 addon 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.

4.22.15 Weblate 3.6

Publicada el 20 de abril de 2019.

- Add support for downloading user data.
- Addons are now automatically triggered upon installation.
- Improved instructions for resolving merge conflicts.
- Cleanup addon 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 addons 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.

4.22.16 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 addon 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.

4.22.17 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.
- Ahora puede configurarse el mensaje de consigna de complemento.
- 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 addon now removes projects for unpaid billings after 45 days.

4.22.18 Weblate 3.4

Publicada el 22 de enero de 2019.

- Added support for XLIFF placeholders.
- 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 addon 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.

4.22.19 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.

4.22.20 Weblate 3.2.2

Publicada el 20 de octubre de 2018.

- Remove no longer needed Babel dependency.
- Updated language definitions.
- Improve documentation for addons, 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.

4.22.21 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.

4.22.22 Weblate 3.2

Publicada el 6 de octubre de 2018.

- Add install_addon management command for automated addon installation.
- Allow more fine grained ratelimit settings.
- Added support for export and import of Excel files.
- Se mejoró la limpieza de componentes cuando se utilizan varios complementos de detección de componentes.
- 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.

4.22.23 Weblate 3.1.1

Publicada el 27 de julio de 2018.

- Fix testsuite failure on some setups.

4.22.24 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 libravator service.
- New addon to mark unchanged translations as needing edit.
- 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.

4.22.25 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.

4.22.26 Weblate 3.0

Publicada el 1.º de junio de 2018.

- Rewritten access control.
- Several code cleanups that lead to moved and renamed modules.
- Nuevo complemento para la detección automática de componentes.
- The `import_project` management command has now slightly different parameters.
- Added basic support for Windows RC files.
- New addon to store contributor names in PO file headers.
- The per component hook scripts are removed, use addons instead.
- Add support for collecting contributor agreements.

- Access control changes are now tracked in history.
- New addon 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.

4.23 Serie 2.x de Weblate

4.23.1 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 addon to configure gettext output wrapping.
- New addon 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.
- Added built-in translation memory.
- 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.

4.23.2 Weblate 2.19.1

Publicada el 20 de febrero de 2018.

- Fixed migration issue on upgrade from 2.18.
- Improved file upload API validation.

4.23.3 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 addons to customize translation workflows.

4.23.4 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.

4.23.5 Weblate 2.17.1

Publicada el 13 de octubre de 2017.

- Fixed running testsuite in some specific situations.
- Locales updates.

4.23.6 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.

4.23.7 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.

4.23.8 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.

4.23.9 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.

4.23.10 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.

4.23.11 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.

4.23.12 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 ALCs are now implemented using Group ACL.
- Added more fine grained privileges control.
- Various minor UI improvements.

4.23.13 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.

4.23.14 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 addon.
- 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.

4.23.15 Weblate 2.10.1

Publicada el 20 de enero de 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

4.23.16 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.

4.23.17 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.

4.23.18 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.

4.23.19 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.

4.23.20 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.

4.23.21 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 not translated 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.

4.23.22 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.

4.23.23 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.

4.23.24 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.

4.23.25 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.

4.23.26 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.

4.24 Serie 1.x de Weblate

4.24.1 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.

4.24.2 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.

4.24.3 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 translating view.

4.24.4 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.

4.24.5 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 libavatar.
- Added possibility to pretty print JSON export.
- Various performance improvements.
- Indicate failing checks or fuzzy strings in progress bars for projects or languages as well.
- Added support for custom pre-commit hooks and committing additional files.
- Rewritten search for better performance and user experience.
- 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.

4.24.6 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.

4.24.7 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 (BB code, 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.

4.24.8 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.
- Caching counts of strings with failing checks.
- Checking of requirements during setup.
- Documentation improvements.

4.24.9 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.

4.24.10 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.

4.25 Serie 0.x de Weblate

4.25.1 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 not translated strings ignores format string only messages.

4.25.2 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.

4.25.3 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.

4.25.4 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.

4.25.5 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).

4.25.6 Weblate 0.4

Publicada el 8 de febrero de 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

4.25.7 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.

4.25.8 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.

4.25.9 Weblate 0.1

Publicada el 6 de febrero de 2012.

- Versión inicial.

W

wlc, [132](#)
wlc.config, [133](#)
wlc.main, [133](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component)/(string:language)/, 106
ANY /, 84	GET /api/components/(string:project)/(string:component)/(string:language)/, 108
/api	GET /api/components/(string:project)/(string:component)/(string:language)/, 108
GET /api/, 86	GET /api/components/(string:project)/(string:component)/(string:language)/, 107
/api/addons	GET /api/components/(string:project)/(string:component)/(string:language)/, 106
GET /api/addons/, 121	GET /api/components/(string:project)/(string:component)/(string:language)/, 110
GET /api/addons/(int:id)/, 121	GET /api/components/(string:project)/(string:component)/(string:language)/, 108
PUT /api/addons/(int:id)/, 122	POST /api/components/(string:project)/(string:component)/(string:language)/, 121
DELETE /api/addons/(int:id)/, 122	POST /api/components/(string:project)/(string:component)/(string:language)/, 110
PATCH /api/addons/(int:id)/, 122	POST /api/components/(string:project)/(string:component)/(string:language)/, 106
/api/changes	DELETE /api/components/(string:project)/(string:component)/(string:language)/, 106
GET /api/changes/, 118	DELETE /api/components/(string:project)/(string:component)/(string:language)/, 110
GET /api/changes/(int:id)/, 118	DELETE /api/components/(string:project)/(string:component)/(string:language)/, 104
/api/component-lists	/api/groups
GET /api/component-lists/, 122	GET /api/groups/, 90
GET /api/component-lists/(str:slug)/, 122	GET /api/groups/(int:id)/, 90
POST /api/component-lists/(str:slug)/components/, 123	POST /api/groups/, 90
PUT /api/component-lists/(str:slug)/, 122	POST /api/groups/(int:id)/componentlists/, 92
DELETE /api/component-lists/(str:slug)/, 123	POST /api/groups/(int:id)/changes/, 92
DELETE /api/component-lists/(str:slug)/components/(string:project)/(string:component)/(string:language)/, 123	POST /api/groups/(int:id)/links/, 92
PATCH /api/component-lists/(str:slug)/, 123	
/api/components	
GET /api/components/, 102	
GET /api/components/(string:project)/(string:component)/(string:language)/, 102	
GET /api/components/(string:project)/(string:component)/(string:language)/, 106	
GET /api/components/(string:project)/(string:component)/(string:language)/, 110	

POST /api/groups/(int:id)/projects/, 92
 POST /api/groups/(int:id)/roles/, 91
 PUT /api/groups/(int:id)/, 91
 DELETE /api/groups/(int:id)/, 91
 DELETE /api/groups/(int:id)/componentlist/(int:component_list_id), 92
 DELETE /api/groups/(int:id)/components/(int:component_id), 92
 DELETE /api/groups/(int:id)/languages/(string:language_code), 92
 DELETE /api/groups/(int:id)/projects/(int:project_id), 92
 PATCH /api/groups/(int:id)/, 91

/api/languages

GET /api/languages/, 94
 GET /api/languages/(string:language)/, 94
 GET /api/languages/(string:language)/statistics/, 95
 POST /api/languages/, 94
 PUT /api/languages/(string:language)/, 95
 DELETE /api/languages/(string:language)/, 95
 PATCH /api/languages/(string:language)/, 95

/api/projects

GET /api/projects/, 96
 GET /api/projects/(string:project)/, 96
 GET /api/projects/(string:project)/changes/, 97
 GET /api/projects/(string:project)/components/, 98
 GET /api/projects/(string:project)/languages/, 101
 GET /api/projects/(string:project)/repository/, 97
 GET /api/projects/(string:project)/statistics/, 101
 POST /api/projects/, 96
 POST /api/projects/(string:project)/components/, 98
 POST /api/projects/(string:project)/repository/, 98
 PUT /api/projects/(string:project)/, 97
 DELETE /api/projects/(string:project)/, 97
 PATCH /api/projects/(string:project)/, 97

/api/roles

GET /api/roles/, 93
 GET /api/roles/(int:id)/, 93

POST /api/roles/, 93
 PUT /api/roles/(int:id)/, 93
 DELETE /api/roles/(int:id)/, 94
 PATCH /api/roles/(int:id)/, 93
 /api/screenshots
 GET /api/screenshots/, 119
 GET /api/screenshots/(int:id)/, 119
 GET /api/screenshots/(int:id)/file/, 119
 POST /api/screenshots/, 120
 POST /api/screenshots/(int:id)/file/, 119
 POST /api/screenshots/(int:id)/units/, 120
 PUT /api/screenshots/(int:id)/, 121
 DELETE /api/screenshots/(int:id)/, 121
 DELETE /api/screenshots/(int:id)/units/(int:unit_id)/, 120
 PATCH /api/screenshots/(int:id)/, 120

/api/tasks

GET /api/tasks/, 123
 GET /api/tasks/(str:uuid)/, 123

/api/translations

GET /api/translations/, 111
 GET /api/translations/(string:project)/(string:component)/, 111
 GET /api/translations/(string:project)/(string:component)/, 113
 GET /api/translations/(string:project)/(string:component)/, 114
 GET /api/translations/(string:project)/(string:component)/, 115
 GET /api/translations/(string:project)/(string:component)/, 116
 GET /api/translations/(string:project)/(string:component)/, 113
 POST /api/translations/(string:project)/(string:component)/, 114
 POST /api/translations/(string:project)/(string:component)/, 115
 POST /api/translations/(string:project)/(string:component)/, 115
 POST /api/translations/(string:project)/(string:component)/, 114
 DELETE /api/translations/(string:project)/(string:component)/, 113

/api/units

GET /api/units/, 116
 GET /api/units/(int:id)/, 116
 PUT /api/units/(int:id)/, 118
 DELETE /api/units/(int:id)/, 118
 PATCH /api/units/(int:id)/, 117

/api/users

GET /api/users/, [87](#)
GET /api/users/(str:username)/, [87](#)
GET /api/users/(str:username)/notifications/,
[89](#)
GET /api/users/(str:username)/notifications/(int:subscription_id)/,
[89](#)
GET /api/users/(str:username)/statistics/,
[89](#)
POST /api/users/, [87](#)
POST /api/users/(str:username)/groups/,
[88](#)
POST /api/users/(str:username)/notifications/,
[89](#)
PUT /api/users/(str:username)/, [88](#)
PUT /api/users/(str:username)/notifications/(int:subscription_id)/,
[89](#)
DELETE /api/users/(str:username)/, [88](#)
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/,
[90](#)
PATCH /api/users/(str:username)/, [88](#)
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/,
[89](#)

/exports

GET /exports/rss/, [127](#)
GET /exports/rss/(string:project)/, [127](#)
GET /exports/rss/(string:project)/(string:component)/,
[127](#)
GET /exports/rss/(string:project)/(string:component)/(string:language)/,
[127](#)
GET /exports/rss/language/(string:language)/,
[127](#)
GET /exports/stats/(string:project)/(string:component)/,
[126](#)

/hooks

GET /hooks/update/(string:project)/,
[124](#)
GET /hooks/update/(string:project)/(string:component)/,
[124](#)
POST /hooks/azure/, [125](#)
POST /hooks/bitbucket/, [124](#)
POST /hooks/gitea/, [125](#)
POST /hooks/gitee/, [125](#)
POST /hooks/github/, [124](#)
POST /hooks/gitlab/, [124](#)
POST /hooks/pagure/, [125](#)

Símbolos

- .XML resource file
 - file format, [71](#)
- add
 - auto_translateopción de línea de comando, [333](#)
- addon ADDON
 - install_addonopción de línea de comando, [339](#)
- age HOURS
 - commit_pendingopción de línea de comando, [334](#)
- author USER@EXAMPLE.COM
 - add_suggestionsopción de línea de comando, [333](#)
- base-file-template TEMPLATE
 - import_projectopción de línea de comando, [337](#)
- check
 - importusersopción de línea de comando, [339](#)
- config PATH
 - wlcopción de línea de comando, [129](#)
- config-section SECTION
 - wlcopción de línea de comando, [129](#)
- configuration CONFIG
 - install_addonopción de línea de comando, [339](#)
- convert
 - wlcopción de línea de comando, [130](#)
- email USER@EXAMPLE.COM
 - createadminopción de línea de comando, [335](#)
- file-format FORMAT
 - import_projectopción de línea de comando, [337](#)
- force
 - loadpoopción de línea de comando, [340](#)
- force-commit
 - pushgitopción de línea de comando, [341](#)
- format {csv,json,text,html}
 - wlcopción de línea de comando, [129](#)
- ignore
 - import_jsonopción de línea de comando, [336](#)
- inconsistent
 - auto_translateopción de línea de comando, [333](#)
- input
 - wlcopción de línea de comando, [130](#)
- key KEY
 - wlcopción de línea de comando, [129](#)
- lang LANGUAGE
 - loadpoopción de línea de comando, [340](#)
- language-code
 - list_translatorsopción de línea de comando, [340](#)
- language-map LANGMAP
 - import_memoryopción de línea de comando, [337](#)
- language-regex REGEX
 - import_projectopción de línea de comando, [337](#)
- license NAME
 - import_projectopción de línea de comando, [338](#)
- license-url URL
 - import_projectopción de línea de comando, [338](#)
- main-component
 - import_projectopción de línea de comando, [337](#)
- main-component COMPONENT
 - import_jsonopción de línea de comando, [336](#)
- mode MODE
 - auto_translateopción de línea de comando, [333](#)
- mt MT
 - auto_translateopción de línea de comando, [333](#)
- name
 - createadminopción de línea de comando, [335](#)

--name-template TEMPLATE
import_project opción de línea de comando, 337

--new-base-template TEMPLATE
import_project opción de línea de comando, 337

--no-password
createadmin opción de línea de comando, 335

--no-privs-update
setupgroups opción de línea de comando, 342

--no-projects-update
setupgroups opción de línea de comando, 342

--no-update
setuplang opción de línea de comando, 342

--output
wlc opción de línea de comando, 130

--overwrite
auto_translate opción de línea de comando, 333
wlc opción de línea de comando, 130

--password PASSWORD
createadmin opción de línea de comando, 335

--project PROJECT
import_json opción de línea de comando, 336

--source PROJECT/COMPONENT
auto_translate opción de línea de comando, 333

--threshold THRESHOLD
auto_translate opción de línea de comando, 333

--update
createadmin opción de línea de comando, 335
import_json opción de línea de comando, 336
install_addon opción de línea de comando, 339

--url URL
wlc opción de línea de comando, 129

--user USERNAME
auto_translate opción de línea de comando, 333

--username USERNAME
createadmin opción de línea de comando, 335

--vcs NAME
import_project opción de línea de comando, 338

A

add_suggestions
weblate admin command, 333

add_suggestions opción de línea de comando
--author USER@EXAMPLE.COM, 333

ADMINS
setting, 177

AKISMET_API_KEY
setting, 289

ALLOWED_HOSTS
setting, 177

Android
file format, 66

ANONYMOUS_USER_NAME
setting, 289

API, 84, 127, 132

Apple strings
file format, 67

ARB
file format, 70

AUDITLOG_EXPIRY
setting, 290

AUTH_LOCK_ATTEMPTS
setting, 290

AUTH_TOKEN_VALID
setting, 291

auto_translate
weblate admin command, 333

auto_translate opción de línea de comando
--add, 333
--inconsistent, 333
--mode MODE, 333
--mt MT, 333
--overwrite, 333
--source PROJECT/COMPONENT, 333
--threshold THRESHOLD, 333
--user USERNAME, 333

AUTO_UPDATE
setting, 290

AUTOFIX_LIST
setting, 291

AVATAR_URL_PREFIX
setting, 290

B

BACKGROUND_TASKS
setting, 292

BASE_DIR
setting, 292

BaseAddon (*clase en weblate.addons.base*), 376

BASIC_LANGUAGES
setting, 292

bilingual
translation, 58

C

can_install() (*método de clase de weblate.addons.base.BaseAddon*), 376

CELERY_BACKUP_OPTIONS, 139

CELERY_BEAT_OPTIONS, 139
 CELERY_MAIN_OPTIONS, 139
 CELERY_MEMORY_OPTIONS, 139
 CELERY_NOTIFY_OPTIONS, 139
 celery_queues
 weblate admin command, 334
 CELERY_TRANSLATE_OPTIONS, 139
 changes
 wlcopción de línea de comando, 130
 CHECK_LIST
 setting, 293
 checkgit
 weblate admin command, 334
 cleanup
 wlcopción de línea de comando, 130
 cleanuptrans
 weblate admin command, 335
 Comma separated values
 file format, 71
 Command (*clase en wlc.main*), 134
 COMMENT_CLEANUP_DAYS
 setting, 293
 commit
 wlcopción de línea de comando, 129
 commit_pending
 weblate admin command, 334
 COMMIT_PENDING_HOURS
 setting, 294
 commit_pendingopción de línea de comando
 --age HOURS, 334
 commitgit
 weblate admin command, 334
 configure() (*método de weblate.addons.base.BaseAddon*), 376
 CONTACT_FORM
 setting, 294
 createadmin
 weblate admin command, 335
 createadminopción de línea de comando
 --email USER@EXAMPLE.COM, 335
 --name, 335
 --no-password, 335
 --password PASSWORD, 335
 --update, 335
 --username USERNAME, 335
 CSP_CONNECT_SRC
 setting, 292
 CSP_FONT_SRC
 setting, 292
 CSP_IMG_SRC
 setting, 292
 CSP_SCRIPT_SRC
 setting, 292
 CSP_STYLE_SRC
 setting, 292
 CSV
 file format, 71

D

daily() (*método de weblate.addons.base.BaseAddon*), 376
 DATA_DIR
 setting, 294
 DATABASE_BACKUP
 setting, 294
 DATABASES
 setting, 177
 DEBUG
 setting, 177
 DEFAULT_ACCESS_CONTROL
 setting, 295
 DEFAULT_ADD_MESSAGE
 setting, 295
 DEFAULT_ADDON_MESSAGE
 setting, 295
 DEFAULT_ADDONS
 setting, 296
 DEFAULT_AUTO_WATCH
 setting, 295
 DEFAULT_COMMIT_MESSAGE
 setting, 295
 DEFAULT_COMMITTER_EMAIL
 setting, 296
 DEFAULT_COMMITTER_NAME
 setting, 296
 DEFAULT_DELETE_MESSAGE
 setting, 295
 DEFAULT_FROM_EMAIL
 setting, 177
 DEFAULT_LANGUAGE
 setting, 296
 DEFAULT_MERGE_MESSAGE
 setting, 295
 DEFAULT_MERGE_STYLE
 setting, 297
 DEFAULT_PULL_MESSAGE
 setting, 297
 DEFAULT_RESTRICTED_COMPONENT
 setting, 295
 DEFAULT_SHARED_TM
 setting, 297
 DEFAULT_TRANSLATION_PROPAGATION
 setting, 297
 download
 wlcopción de línea de comando, 130
 DTD
 file format, 73
 dump_memory
 weblate admin command, 335
 dumpuserdata
 weblate admin command, 335

E

ENABLE_AVATARS
 setting, 297
 ENABLE_HOOKS

setting, 297
ENABLE_HTTPS
setting, 298
ENABLE_SHARING
setting, 298

F

file format
.XML resource file, 71
Android, 66
Apple strings, 67
ARB, 70
Comma separated values, 71
CSV, 71
DTD, 73
gettext, 60
go-i18n, 69
GWT properties, 64
i18next, 69
INI translations, 64, 65
Java properties, 63
Joomla translations, 65
JSON, 68
PHP strings, 67
PO, 60
Qt, 65
RC, 74
RESX, 71
Ruby YAML, 72
Ruby YAML Ain't Markup Language, 72
string resources, 66
TS, 65
XLIFF, 62
XML, 73
YAML, 72
YAML Ain't Markup Language, 72

G

get() (método de *wlc.Weblate*), 133
get_add_form() (método de clase de *weblate.addons.base.BaseAddon*), 376
GET_HELP_URL
setting, 298
get_settings_form() (método de *weblate.addons.base.BaseAddon*), 376
gettext
file format, 60
GITHUB_CREDENTIALS
setting, 299
GITHUB_TOKEN
setting, 299
GITHUB_USERNAME
setting, 299
GITLAB_CREDENTIALS
setting, 298
GITLAB_TOKEN
setting, 299
GITLAB_USERNAME

setting, 298
go-i18n
file format, 69
GOOGLE_ANALYTICS_ID
setting, 299
GWT properties
file format, 64

H

HIDE_REPO_CREDENTIALS
setting, 300
HIDE_VERSION
setting, 300

I

i18next
file format, 69
import_demo
weblate admin command, 336
import_json
weblate admin command, 336
import_jsonopción de línea de comando
--ignore, 336
--main-component COMPONENT, 336
--project PROJECT, 336
--update, 336
import_memory
weblate admin command, 337
import_memoryopción de línea de comando
--language-map LANGMAP, 337
import_project
weblate admin command, 337
import_projectopción de línea de comando
--base-file-template TEMPLATE, 337
--file-format FORMAT, 337
--language-regex REGEX, 337
--license NAME, 338
--license-url URL, 338
--main-component, 337
--name-template TEMPLATE, 337
--new-base-template TEMPLATE, 337
--vcs NAME, 338
importuserdata
weblate admin command, 339
importusers
weblate admin command, 339
importusersopción de línea de comando
--check, 339
INI translations
file format, 64, 65
install_addon
weblate admin command, 339
install_addonopción de línea de comando
--addon ADDON, 339
--configuration CONFIG, 339

- update, 339
- IP_BEHIND_REVERSE_PROXY
 - setting, 300
- IP_PROXY_HEADER
 - setting, 300
- IP_PROXY_OFFSET
 - setting, 301
- iPad
 - translation, 67
- iPhone
 - translation, 67

J

- Java properties
 - file format, 63
- Joomla translations
 - file format, 65
- JSON
 - file format, 68

L

- LEGAL_URL
 - setting, 301
- LICENSE_EXTRA
 - setting, 301
- LICENSE_FILTER
 - setting, 302
- LICENSE_REQUIRED
 - setting, 302
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
 - setting, 302
- list_languages
 - weblate admin command, 340
- list_translators
 - weblate admin command, 340
- list_translatorsopción de línea de comando
 - language-code, 340
- list_versions
 - weblate admin command, 340
- list-components
 - wlcopción de línea de comando, 129
- list-languages
 - wlcopción de línea de comando, 129
- list-projects
 - wlcopción de línea de comando, 129
- list-translations
 - wlcopción de línea de comando, 129
- load() (*método de wlc.config.WeblateConfig*), 133
- loadpo
 - weblate admin command, 340
- loadpoopción de línea de comando
 - force, 340
 - lang LANGUAGE, 340
- LOCALIZE_CDN_PATH
 - setting, 302
- LOCALIZE_CDN_URL
 - setting, 302

- lock
 - wlcopción de línea de comando, 130
- lock_translation
 - weblate admin command, 341
- lock-status
 - wlcopción de línea de comando, 130
- LOGIN_REQUIRED_URLS
 - setting, 303
- LOGIN_REQUIRED_URLS_EXCEPTIONS
 - setting, 303
- ls
 - wlcopción de línea de comando, 129

M

- MACHINE_TRANSLATION_SERVICES
 - setting, 304
- main() (*en el módulo wlc.main*), 133
- MATOMO_SITE_ID
 - setting, 303
- MATOMO_URL
 - setting, 304
- monolingual
 - translation, 58
- move_language
 - weblate admin command, 341
- MT_APERTIUM_APY
 - setting, 304
- MT_AWS_ACCESS_KEY_ID
 - setting, 305
- MT_AWS_REGION
 - setting, 305
- MT_AWS_SECRET_ACCESS_KEY
 - setting, 305
- MT_BAIDU_ID
 - setting, 305
- MT_BAIDU_SECRET
 - setting, 305
- MT_DEEPL_API_VERSION
 - setting, 305
- MT_DEEPL_KEY
 - setting, 306
- MT_GOOGLE_CREDENTIALS
 - setting, 306
- MT_GOOGLE_KEY
 - setting, 306
- MT_GOOGLE_LOCATION
 - setting, 306
- MT_GOOGLE_PROJECT
 - setting, 306
- MT_MICROSOFT_BASE_URL
 - setting, 306
- MT_MICROSOFT_COGNITIVE_KEY
 - setting, 306
- MT_MICROSOFT_ENDPOINT_URL
 - setting, 307
- MT_MICROSOFT_REGION
 - setting, 307
- MT_MODERNMT_KEY

setting, 307
MT_MODERNMT_URL
 setting, 307
MT_MYMEMORY_EMAIL
 setting, 307
MT_MYMEMORY_KEY
 setting, 307
MT_MYMEMORY_USER
 setting, 307
MT_NETEASE_KEY
 setting, 308
MT_NETEASE_SECRET
 setting, 308
MT_SAP_BASE_URL
 setting, 308
MT_SAP_PASSWORD
 setting, 309
MT_SAP_SANDBOX_APIKEY
 setting, 309
MT_SAP_USE_MT
 setting, 309
MT_SAP_USERNAME
 setting, 309
MT_SERVICES
 setting, 304
MT_TMSERVER
 setting, 308
MT_YANDEX_KEY
 setting, 308
MT_YOUDAO_ID
 setting, 308
MT_YOUDAO_SECRET
 setting, 308
módulo
 wlc, 132
 wlc.config, 133
 wlc.main, 133

N

NEARBY_MESSAGES
 setting, 309

P

PAGURE_CREDENTIALS
 setting, 309
PAGURE_TOKEN
 setting, 310
PAGURE_USERNAME
 setting, 310
PHP strings
 file format, 67
PIWIK_SITE_ID
 setting, 303
PIWIK_URL
 setting, 304
PO
 file format, 60
post() (método de wlc.Weblate), 133

post_add() (método de weblate.addons.base.BaseAddon), 376
post_commit() (método de weblate.addons.base.BaseAddon), 376
post_push() (método de weblate.addons.base.BaseAddon), 376
post_update() (método de weblate.addons.base.BaseAddon), 376
pre_commit() (método de weblate.addons.base.BaseAddon), 376
pre_push() (método de weblate.addons.base.BaseAddon), 376
pre_update() (método de weblate.addons.base.BaseAddon), 376
pull
 wlcopción de línea de comando, 129
push
 wlcopción de línea de comando, 130
pushgit
 weblate admin command, 341
pushgitopción de línea de comando
 --force-commit, 341
Python, 132

Q

Qt
 file format, 65

R

RATELIMIT_ATTEMPTS
 setting, 310
RATELIMIT_LOCKOUT
 setting, 310
RATELIMIT_WINDOW
 setting, 310
RC
 file format, 74
register_command() (en el módulo wlc.main), 133
REGISTRATION_ALLOW_BACKENDS
 setting, 311
REGISTRATION_CAPTCHA
 setting, 311
REGISTRATION_EMAIL_MATCH
 setting, 311
REGISTRATION_OPEN
 setting, 311
repo
 wlcopción de línea de comando, 130
REPOSITORY_ALERT_THRESHOLD
 setting, 312
REQUIRE_LOGIN
 setting, 312
reset
 wlcopción de línea de comando, 130
REST, 84
RESX
 file format, 71
RFC

RFC 4646, 57
 Ruby YAML
 file format, 72
 Ruby YAML Ain't Markup Language
 file format, 72

S

save_state() (método de *weblate.addons.base.BaseAddon*), 376
 SECRET_KEY
 setting, 177
 SENTRY_DSN
 setting, 312
 SERVER_EMAIL
 setting, 178
 SESSION_COOKIE_AGE_AUTHENTICATED
 setting, 312
 SESSION_ENGINE
 setting, 177
 setting
 ADMINS, 177
 AKISMET_API_KEY, 289
 ALLOWED_HOSTS, 177
 ANONYMOUS_USER_NAME, 289
 AUDITLOG_EXPIRY, 290
 AUTH_LOCK_ATTEMPTS, 290
 AUTH_TOKEN_VALID, 291
 AUTO_UPDATE, 290
 AUTOFIX_LIST, 291
 AVATAR_URL_PREFIX, 290
 BACKGROUND_TASKS, 292
 BASE_DIR, 292
 BASIC_LANGUAGES, 292
 CHECK_LIST, 293
 COMMENT_CLEANUP_DAYS, 293
 COMMIT_PENDING_HOURS, 294
 CONTACT_FORM, 294
 CSP_CONNECT_SRC, 292
 CSP_FONT_SRC, 292
 CSP_IMG_SRC, 292
 CSP_SCRIPT_SRC, 292
 CSP_STYLE_SRC, 292
 DATA_DIR, 294
 DATABASE_BACKUP, 294
 DATABASES, 177
 DEBUG, 177
 DEFAULT_ACCESS_CONTROL, 295
 DEFAULT_ADD_MESSAGE, 295
 DEFAULT_ADDON_MESSAGE, 295
 DEFAULT_ADDONS, 296
 DEFAULT_AUTO_WATCH, 295
 DEFAULT_COMMIT_MESSAGE, 295
 DEFAULT_COMMITER_EMAIL, 296
 DEFAULT_COMMITER_NAME, 296
 DEFAULT_DELETE_MESSAGE, 295
 DEFAULT_FROM_EMAIL, 177
 DEFAULT_LANGUAGE, 296
 DEFAULT_MERGE_MESSAGE, 295

DEFAULT_MERGE_STYLE, 297
 DEFAULT_PULL_MESSAGE, 297
 DEFAULT_RESTRICTED_COMPONENT, 295
 DEFAULT_SHARED_TM, 297
 DEFAULT_TRANSLATION_PROPAGATION, 297
 ENABLE_AVATARS, 297
 ENABLE_HOOKS, 297
 ENABLE_HTTPS, 298
 ENABLE_SHARING, 298
 GET_HELP_URL, 298
 GITHUB_CREDENTIALS, 299
 GITHUB_TOKEN, 299
 GITHUB_USERNAME, 299
 GITLAB_CREDENTIALS, 298
 GITLAB_TOKEN, 299
 GITLAB_USERNAME, 298
 GOOGLE_ANALYTICS_ID, 299
 HIDE_REPO_CREDENTIALS, 300
 HIDE_VERSION, 300
 IP_BEHIND_REVERSE_PROXY, 300
 IP_PROXY_HEADER, 300
 IP_PROXY_OFFSET, 301
 LEGAL_URL, 301
 LICENSE_EXTRA, 301
 LICENSE_FILTER, 302
 LICENSE_REQUIRED, 302
 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 302
 LOCALIZE_CDN_PATH, 302
 LOCALIZE_CDN_URL, 302
 LOGIN_REQUIRED_URLS, 303
 LOGIN_REQUIRED_URLS_EXCEPTIONS, 303
 MACHINE_TRANSLATION_SERVICES, 304
 MATOMO_SITE_ID, 303
 MATOMO_URL, 304
 MT_APERTIUM_API_KEY, 304
 MT_AWS_ACCESS_KEY_ID, 305
 MT_AWS_REGION, 305
 MT_AWS_SECRET_ACCESS_KEY, 305
 MT_BAIDU_ID, 305
 MT_BAIDU_SECRET, 305
 MT_DEEPL_API_VERSION, 305
 MT_DEEPL_KEY, 306
 MT_GOOGLE_CREDENTIALS, 306
 MT_GOOGLE_KEY, 306
 MT_GOOGLE_LOCATION, 306
 MT_GOOGLE_PROJECT, 306
 MT_MICROSOFT_BASE_URL, 306
 MT_MICROSOFT_COGNITIVE_KEY, 306
 MT_MICROSOFT_ENDPOINT_URL, 307
 MT_MICROSOFT_REGION, 307
 MT_MODERNMT_KEY, 307
 MT_MODERNMT_URL, 307
 MT_MYMEMORY_EMAIL, 307
 MT_MYMEMORY_KEY, 307
 MT_MYMEMORY_USER, 307
 MT_NETEASE_KEY, 308

- MT_NETEASE_SECRET, 308
 - MT_SAP_BASE_URL, 308
 - MT_SAP_PASSWORD, 309
 - MT_SAP_SANDBOX_APIKEY, 309
 - MT_SAP_USE_MT, 309
 - MT_SAP_USERNAME, 309
 - MT_SERVICES, 304
 - MT_TMSERVER, 308
 - MT_YANDEX_KEY, 308
 - MT_YOUDAO_ID, 308
 - MT_YOUDAO_SECRET, 308
 - NEARBY_MESSAGES, 309
 - PAGURE_CREDENTIALS, 309
 - PAGURE_TOKEN, 310
 - PAGURE_USERNAME, 310
 - PIWIK_SITE_ID, 303
 - PIWIK_URL, 304
 - RATELIMIT_ATTEMPTS, 310
 - RATELIMIT_LOCKOUT, 310
 - RATELIMIT_WINDOW, 310
 - REGISTRATION_ALLOW_BACKENDS, 311
 - REGISTRATION_CAPTCHA, 311
 - REGISTRATION_EMAIL_MATCH, 311
 - REGISTRATION_OPEN, 311
 - REPOSITORY_ALERT_THRESHOLD, 312
 - REQUIRE_LOGIN, 312
 - SECRET_KEY, 177
 - SENTRY_DSN, 312
 - SERVER_EMAIL, 178
 - SESSION_COOKIE_AGE_AUTHENTICATED, 312
 - SESSION_ENGINE, 177
 - SIMPLIFY_LANGUAGES, 312
 - SINGLE_PROJECT, 313
 - SITE_DOMAIN, 313
 - SITE_TITLE, 313
 - SPECIAL_CHARS, 313
 - STATUS_URL, 314
 - SUGGESTION_CLEANUP_DAYS, 314
 - UPDATE_LANGUAGES, 314
 - URL_PREFIX, 314
 - VCS_BACKENDS, 314
 - VCS_CLONE_DEPTH, 315
 - WEBLATE_ADDONS, 315
 - WEBLATE_EXPORTERS, 316
 - WEBLATE_FORMATS, 316
 - WEBLATE_GPG_IDENTITY, 316
 - WEBSITE_REQUIRED, 316
 - setupgroups
 - weblate admin command, 342
 - setupgroupsopción de línea de comando
 - no-privs-update, 342
 - no-projects-update, 342
 - setuplang
 - weblate admin command, 342
 - setuplangopción de línea de comando
 - no-update, 342
 - show
 - wlcopción de línea de comando, 129
 - SIMPLIFY_LANGUAGES
 - setting, 312
 - SINGLE_PROJECT
 - setting, 313
 - SITE_DOMAIN
 - setting, 313
 - SITE_TITLE
 - setting, 313
 - SPECIAL_CHARS
 - setting, 313
 - statistics
 - wlcopción de línea de comando, 130
 - STATUS_URL
 - setting, 314
 - stay_on_create (atributo de *weblate.addons.base.BaseAddon*), 376
 - store_post_load() (método de *weblate.addons.base.BaseAddon*), 377
 - string resources
 - file format, 66
 - SUGGESTION_CLEANUP_DAYS
 - setting, 314
- ## T
- translation
 - bilingual, 58
 - iPad, 67
 - iPhone, 67
 - monolingual, 58
 - TS
 - file format, 65
- ## U
- unit_pre_create() (método de *weblate.addons.base.BaseAddon*), 377
 - unlock
 - wlcopción de línea de comando, 130
 - unlock_translation
 - weblate admin command, 341
 - UPDATE_LANGUAGES
 - setting, 314
 - updatechecks
 - weblate admin command, 342
 - updategit
 - weblate admin command, 342
 - upload
 - wlcopción de línea de comando, 130
 - URL_PREFIX
 - setting, 314
 - UWSGI_WORKERS, 139
- ## V
- variables de entorno
 - CELERY_BACKUP_OPTIONS, 139, 153
 - CELERY_BEAT_OPTIONS, 139, 153
 - CELERY_MAIN_OPTIONS, 139, 153
 - CELERY_MEMORY_OPTIONS, 139, 153

- CELERY_NOTIFY_OPTIONS, 139, 153
- CELERY_TRANSLATE_OPTIONS, 139, 153
- POSTGRES_ALTER_ROLE, 149
- POSTGRES_DATABASE, 149
- POSTGRES_HOST, 149
- POSTGRES_PASSWORD, 149
- POSTGRES_PORT, 149
- POSTGRES_SSL_MODE, 149
- POSTGRES_USER, 149
- REDIS_DB, 150
- REDIS_HOST, 150
- REDIS_PASSWORD, 150
- REDIS_PORT, 150
- REDIS_TLS, 150
- REDIS_VERIFY_SSL, 150
- ROLLBAR_ENVIRONMENT, 152
- ROLLBAR_KEY, 152
- SENTRY_DSN, 152
- SENTRY_ENVIRONMENT, 152
- SOCIAL_AUTH_SLACK_SECRET, 149
- UWSGI_WORKERS, 139, 153
- WEBLATE_ADD_ADDONS, 153
- WEBLATE_ADD_APPS, 152
- WEBLATE_ADD_AUTOFIX, 152
- WEBLATE_ADD_CHECK, 152
- WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 142
- WEBLATE_ADMIN_EMAIL, 139, 140, 145
- WEBLATE_ADMIN_NAME, 139, 140
- WEBLATE_ADMIN_PASSWORD, 136, 139, 140
- WEBLATE_AKISMET_API_KEY, 143, 349
- WEBLATE_ALLOWED_HOSTS, 141, 177, 181, 182, 313
- WEBLATE_AUTH_LDAP_BIND_DN, 146
- WEBLATE_AUTH_LDAP_BIND_PASSWORD, 146
- WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERENCE, 146
- WEBLATE_AUTH_LDAP_SERVER_URI, 146
- WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 146
- WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 146
- WEBLATE_AUTH_LDAP_USER_SEARCH, 146
- WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 146
- WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 146
- WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELETE, 146
- WEBLATE_BASIC_LANGUAGES, 144
- WEBLATE_CONTACT_FORM, 141
- WEBLATE_CSP_CONNECT_SRC, 144
- WEBLATE_CSP_FONT_SRC, 144
- WEBLATE_CSP_IMG_SRC, 144
- WEBLATE_CSP_SCRIPT_SRC, 144
- WEBLATE_CSP_STYLE_SRC, 144
- WEBLATE_DATABASE_BACKUP, 150
- WEBLATE_DEBUG, 140
- WEBLATE_DEFAULT_ACCESS_CONTROL, 143
- WEBLATE_DEFAULT_AUTO_WATCH, 144
- WEBLATE_DEFAULT_COMMITTER_EMAIL, 143
- WEBLATE_DEFAULT_COMMITTER_NAME, 143
- WEBLATE_DEFAULT_FROM_EMAIL, 140
- WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 143
- WEBLATE_DEFAULT_SHARED_TM, 143
- WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 143
- WEBLATE_EMAIL_BACKEND, 151
- WEBLATE_EMAIL_HOST, 150
- WEBLATE_EMAIL_HOST_PASSWORD, 151
- WEBLATE_EMAIL_HOST_USER, 151
- WEBLATE_EMAIL_PORT, 151
- WEBLATE_EMAIL_USE_SSL, 151
- WEBLATE_EMAIL_USE_TLS, 151
- WEBLATE_ENABLE_HTTPS, 141
- WEBLATE_GET_HELP_URL, 151
- WEBLATE_GITHUB_TOKEN, 143
- WEBLATE_GITHUB_USERNAME, 142
- WEBLATE_GITLAB_TOKEN, 143
- WEBLATE_GITLAB_USERNAME, 143
- WEBLATE_GOOGLE_ANALYTICS_ID, 142
- WEBLATE_GPG_IDENTITY, 143
- WEBLATE_HIDE_VERSION, 144
- WEBLATE_IP_PROXY_HEADER, 142
- WEBLATE_LEGAL_URL, 151
- WEBLATE_LICENSE_FILTER, 144
- WEBLATE_LICENSE_REQUIRED, 144
- WEBLATE_LOCALIZE_CDN_PATH, 152
- WEBLATE_LOCALIZE_CDN_URL, 152
- WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 142
- WEBLATE_LOGLEVEL, 140
- WEBLATE_MT_APERTIUM_API_KEY, 145
- WEBLATE_MT_AWS_ACCESS_KEY_ID, 145
- WEBLATE_MT_AWS_REGION, 145
- WEBLATE_MT_AWS_SECRET_ACCESS_KEY, 145
- WEBLATE_MT_DEEPL_API_VERSION, 145
- WEBLATE_MT_DEEPL_KEY, 145
- WEBLATE_MT_GLOSBE_ENABLED, 145
- WEBLATE_MT_GOOGLE_KEY, 145
- WEBLATE_MT_MICROSOFT_BASE_URL, 145
- WEBLATE_MT_MICROSOFT_COGNITIVE_KEY, 145
- WEBLATE_MT_MICROSOFT_ENDPOINT_URL, 145
- WEBLATE_MT_MICROSOFT_REGION, 145
- WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED, 145
- WEBLATE_MT_MODERNMT_KEY, 145
- WEBLATE_MT_MYMOMORY_ENABLED, 145
- WEBLATE_MT_SAP_BASE_URL, 145
- WEBLATE_MT_SAP_PASSWORD, 146
- WEBLATE_MT_SAP_SANDBOX_APIKEY, 145

WEBLATE_MT_SAP_USE_MT, 146
 WEBLATE_MT_SAP_USERNAME, 145
 WEBLATE_NO_EMAIL_AUTH, 149
 WEBLATE_PAGURE_TOKEN, 143
 WEBLATE_PAGURE_USERNAME, 143
 WEBLATE_RATELIMIT_ATTEMPTS, 144, 351
 WEBLATE_RATELIMIT_LOCKOUT, 144
 WEBLATE_RATELIMIT_WINDOW, 144
 WEBLATE_REGISTRATION_ALLOW_BACKENDS, 141
 WEBLATE_REGISTRATION_OPEN, 141
 WEBLATE_REMOVE_ADDONS, 153
 WEBLATE_REMOVE_APPS, 152
 WEBLATE_REMOVE_AUTOFIX, 152
 WEBLATE_REMOVE_CHECK, 152
 WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTED, 142
 WEBLATE_REQUIRE_LOGIN, 142, 312
 WEBLATE_SAML_IDP_ENTITY_ID, 149
 WEBLATE_SAML_IDP_URL, 149
 WEBLATE_SAML_IDP_X509CERT, 149
 WEBLATE_SECURE_PROXY_SSL_HEADER, 142
 WEBLATE_SERVER_EMAIL, 140
 WEBLATE_SERVICE, 139, 153
 WEBLATE_SILENCED_SYSTEM_CHECKS, 144, 204
 WEBLATE_SIMPLIFY_LANGUAGES, 143
 WEBLATE_SITE_DOMAIN, 140, 179, 197, 313
 WEBLATE_SITE_TITLE, 140
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 148
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 148
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 148
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, 148
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID, 148
 WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 147
 WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 147
 WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 147
 WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 147
 WEBLATE_SOCIAL_AUTH_FEDORA, 148
 WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 147
 WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 147
 WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 148
 WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 148
 WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 148
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 147
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 147
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 147
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 147
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 148
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 148
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 148
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 148
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 148
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 148
 WEBLATE_SOCIAL_AUTH_OPENSUSE, 148
 WEBLATE_SOCIAL_AUTH_SLACK_KEY, 149
 WEBLATE_SOCIAL_AUTH_UBUNTU, 148
 WEBLATE_STATUS_URL, 151
 WEBLATE_TIME_ZONE, 141
 WEBLATE_URL_PREFIX, 144
 WEBLATE_WEBSITE_REQUIRED, 144
 WL_BRANCH, 286
 WL_COMPONENT_NAME, 287
 WL_COMPONENT_SLUG, 286
 WL_COMPONENT_URL, 287
 WL_ENGAGE_URL, 287
 WL_FILE_FORMAT, 286
 WL_FILEMASK, 286
 WL_LANGUAGE, 286
 WL_NEW_BASE, 286
 WL_PATH, 286
 WL_PROJECT_SLUG, 286
 WL_PROJECT_SLUG, 287
 WL_REPO, 286
 WL_TEMPLATE, 286
 WL_VCS, 286
 VCS_BACKENDS setting, 314
 VCS_CLONE_DEPTH setting, 315
 version
 wlcopción de línea de comando, 129

W

Weblate (*clase en wlc*), 133
 weblate admin command
 add_suggestions, 333
 auto_translate, 333
 celery_queues, 334
 checkgit, 334
 cleanuptrans, 335
 commit_pending, 334

- commitgit, 334
- createadmin, 335
- dump_memory, 335
- dumpuserdata, 335
- import_demo, 336
- import_json, 336
- import_memory, 337
- import_project, 337
- importuserdata, 339
- importusers, 339
- install_addon, 339
- list_languages, 340
- list_translators, 340
- list_versions, 340
- loadpo, 340
- lock_translation, 341
- move_language, 341
- pushgit, 341
- setupgroups, 342
- setuplang, 342
- unlock_translation, 341
- updatechecks, 342
- updategit, 342
- WEBLATE_ADDONS
 - setting, 315
- WEBLATE_ADMIN_EMAIL, 139, 140, 145
- WEBLATE_ADMIN_NAME, 139, 140
- WEBLATE_ADMIN_PASSWORD, 136, 139, 140
- WEBLATE_AKISMET_API_KEY, 349
- WEBLATE_ALLOWED_HOSTS, 177, 181, 182, 313
- WEBLATE_EMAIL_PORT, 151
- WEBLATE_EMAIL_USE_SSL, 151
- WEBLATE_EMAIL_USE_TLS, 151
- WEBLATE_EXPORTERS
 - setting, 316
- WEBLATE_FORMATS
 - setting, 316
- WEBLATE_GPG_IDENTITY
 - setting, 316
- WEBLATE_LOCALIZE_CDN_PATH, 152
- WEBLATE_RATELIMIT_ATTEMPTS, 351
- WEBLATE_REQUIRE_LOGIN, 312
- WEBLATE_SECURE_PROXY_SSL_HEADER, 142
- WEBLATE_SERVICE, 139
- WEBLATE_SILENCED_SYSTEM_CHECKS, 204
- WEBLATE_SITE_DOMAIN, 179, 197, 313
- WeblateConfig (*clase en wlc.config*), 133
- WeblateException, 132
- WEBSITE_REQUIRED
 - setting, 316
- wlc, 127
 - módulo, 132
- wlc.config
 - módulo, 133
- wlc.main
 - módulo, 133
- wlcopción de línea de comando
 - config PATH, 129

- config-section SECTION, 129
- convert, 130
- format {csv,json,text,html}, 129
- input, 130
- key KEY, 129
- output, 130
- overwrite, 130
- url URL, 129
- changes, 130
- cleanup, 130
- commit, 129
- download, 130
- list-components, 129
- list-languages, 129
- list-projects, 129
- list-translations, 129
- lock, 130
- lock-status, 130
- ls, 129
- pull, 129
- push, 130
- repo, 130
- reset, 130
- show, 129
- statistics, 130
- unlock, 130
- upload, 130
- version, 129

X

- XLIFF
 - file format, 62
- XML
 - file format, 73

Y

- YAML
 - file format, 72
- YAML Ain't Markup Language
 - file format, 72