



The Weblate Manual

Versión 4.12.1

Michal Čihař

29 de abril de 2022

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	21
1.5	Glosario	25
1.6	Comprobaciones y correcciones	27
1.7	Búsquedas	50
1.8	Flujos de trabajo de traducción	56
1.9	Preguntas frecuentes	59
1.10	Formatos de archivo admitidos	67
1.11	Integración de control de versiones	88
1.12	API REST de Weblate	96
1.13	Cliente de Weblate	143
1.14	API de Python de Weblate	148
2	Documentación para administradores	150
2.1	Instrucciones de configuración	150
2.2	Implantaciones de Weblate	215
2.3	Actualizar Weblate	216
2.4	Respaldar y trasladar Weblate	224
2.5	Autenticación	230
2.6	Control de acceso	241
2.7	Proyectos de traducción	250
2.8	Definiciones de idioma	268
2.9	Regionalización continua	271
2.10	Licensing translations	280
2.11	Proceso de traducción	281
2.12	Comprobaciones y correcciones	287
2.13	Traducción automática	296
2.14	Complementos	303
2.15	Memoria de traducción	319
2.16	Configuración	321
2.17	Configuración de muestra	352
2.18	Órdenes de gestión	368
2.19	Anuncios	379
2.20	Listas de componentes	382
2.21	Módulos opcionales de Weblate	383
2.22	Personalizar Weblate	388
2.23	Interfaz de gestión	390
2.24	Obtener ayuda con Weblate	398

2.25	Documentos legales	401
3	Documentación para contribuidores	403
3.1	Contribuir con Weblate	403
3.2	Empezar a contribuir con código a Weblate	405
3.3	Código fuente de Weblate	409
3.4	Depurar Weblate	410
3.5	Weblate por dentro	412
3.6	Developing add-ons	413
3.7	Interfaz de Weblate	415
3.8	Informar de problemas en Weblate	416
3.9	Weblate testsuite and continuous integration	417
3.10	Esquemas de datos	418
3.11	Releasing Weblate	422
3.12	Seguridad y privacidad	423
3.13	Contributing to Weblate modules	423
3.14	Acerca de Weblate	424
3.15	Licencia	425
4	Historial de cambios	426
4.1	Weblate 4.12.1	426
4.2	Weblate 4.12	426
4.3	Weblate 4.11.2	427
4.4	Weblate 4.11.1	427
4.5	Weblate 4.11	427
4.6	Weblate 4.10.1	428
4.7	Weblate 4.10	428
4.8	Weblate 4.9.1	429
4.9	Weblate 4.9	429
4.10	Weblate 4.8.1	430
4.11	Weblate 4.8	430
4.12	Weblate 4.7.2	431
4.13	Weblate 4.7.1	431
4.14	Weblate 4.7	431
4.15	Weblate 4.6.2	432
4.16	Weblate 4.6.1	432
4.17	Weblate 4.6	432
4.18	Weblate 4.5.3	433
4.19	Weblate 4.5.2	433
4.20	Weblate 4.5.1	434
4.21	Weblate 4.5	434
4.22	Weblate 4.4.2	435
4.23	Weblate 4.4.1	435
4.24	Weblate 4.4	435
4.25	Weblate 4.3.2	436
4.26	Weblate 4.3.1	437
4.27	Weblate 4.3	437
4.28	Weblate 4.2.2	438
4.29	Weblate 4.2.1	438
4.30	Weblate 4.2	438
4.31	Weblate 4.1.1	439
4.32	Weblate 4.1	439
4.33	Weblate 4.0.4	441
4.34	Weblate 4.0.3	441
4.35	Weblate 4.0.2	441
4.36	Weblate 4.0.1	442
4.37	Weblate 4.0	442
4.38	Serie 3.x de Weblate	443

4.39	Serie 2.x de Weblate	454
4.40	Serie 1.x de Weblate	465
4.41	Serie 0.x de Weblate	469
Índice de Módulos Python		473
HTTP Routing Table		474
Índice		477

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.

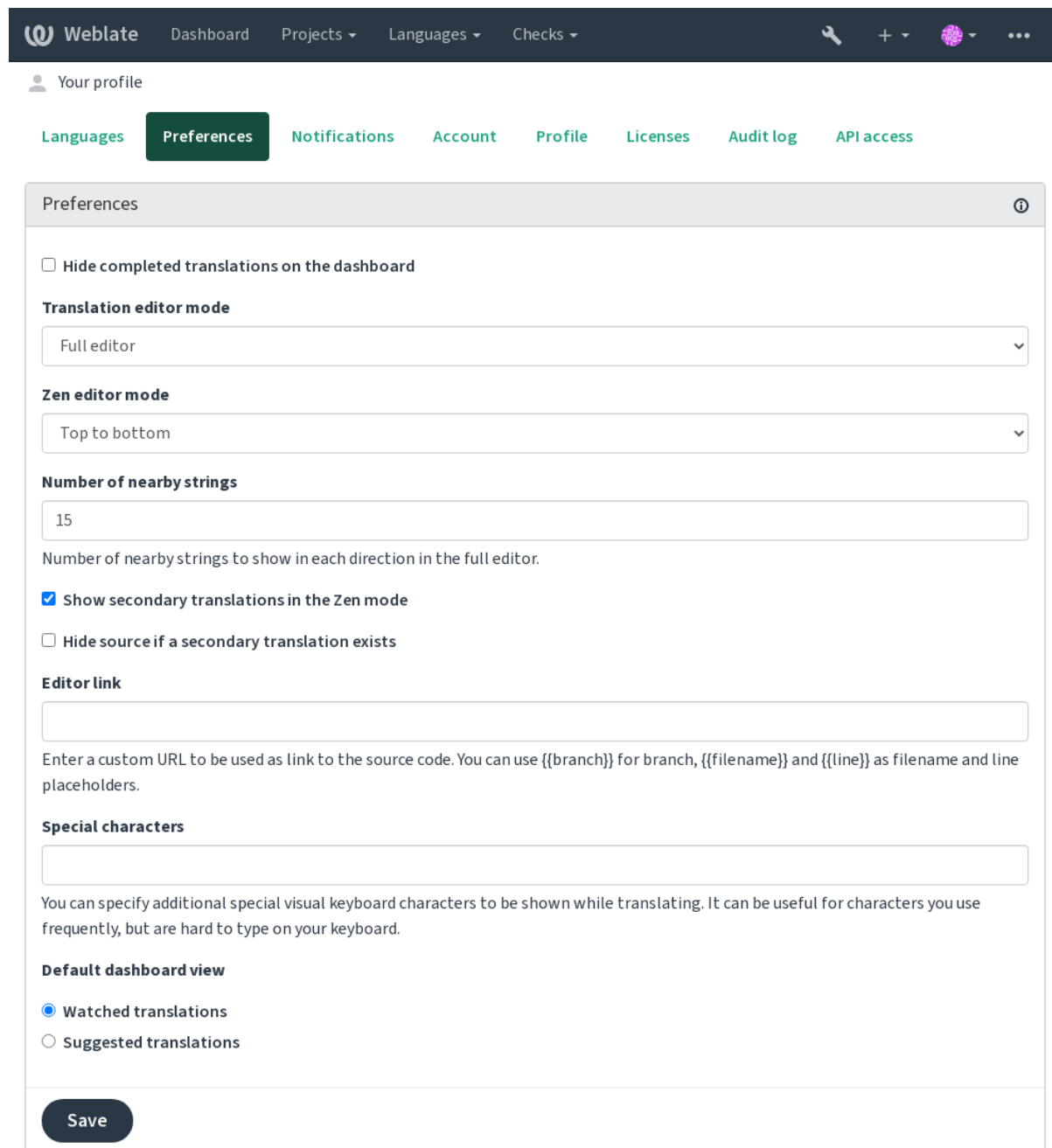
1.2.2 Cuadro de mando

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, '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 series of tabs: 'Languages', 'Preferences' (which is active and highlighted in dark green), 'Notifications', 'Account', 'Profile', 'Licenses', 'Audit log', and 'API access'. The main content area is titled 'Preferences' and contains several settings sections: 'Hide completed translations on the dashboard' (unchecked), 'Translation editor mode' (set to 'Full editor'), 'Zen editor mode' (set to 'Top to bottom'), 'Number of nearby strings' (set to '15' with a description: 'Number of nearby strings to show in each direction in the full editor.'), 'Show secondary translations in the Zen mode' (checked), 'Hide source if a secondary translation exists' (unchecked), 'Editor link' (empty text field with a description: 'Enter a custom URL to be used as link to the source code. You can use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders.'), 'Special characters' (empty text field with a description: 'You can specify additional special visual keyboard characters to be shown while translating. It can be useful for characters you use frequently, but are hard to type on your keyboard.'), and 'Default dashboard view' (with radio buttons for 'Watched translations' (selected) and 'Suggested translations'). A 'Save' button is at the bottom of the preferences panel.

Powered by Weblate 4.12 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.


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

 Dashboard

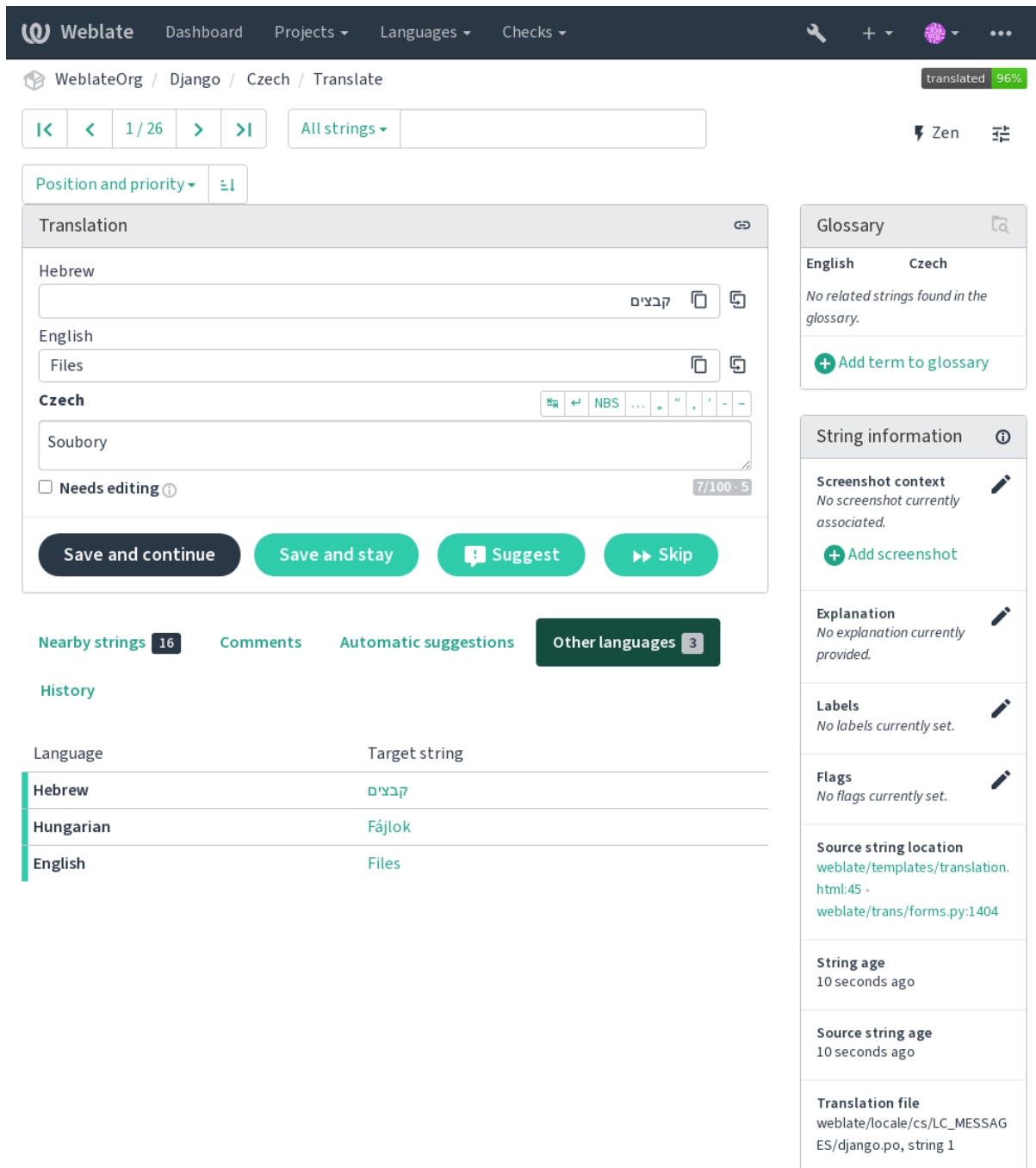
[Watched translations 13](#)
[Suggested translations 5](#)
[Insights ▾](#)
[Search](#)

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

Powered by Weblate 4.12
 [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:



Webblate Dashboard Projects Languages Checks

WebblateOrg / Django / Czech / Translate translated 96%

1 / 26 All strings

Position and priority

Translation

Hebrew

English

Files

Czech

Soubory

Needs editing 7/100 · 5

Save and continue Save and stay Suggest Skip

Nearby strings 16 Comments Automatic suggestions Other languages 3

History

Language	Target string
Hebrew	קבצים
Hungarian	Fájlok
English	Files

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
No flags currently set.

Source string location
weblate/templates/translation.html:45
weblate/trans/forms.py:1404

String age
10 seconds ago

Source string age
10 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](#).

Caracteres especiales

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

1.2.6 Notificaciones

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

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

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

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

Nota: No recibirá ninguna notificación por sus propias acciones.

Web

late

Dashboard

Projects

Languages

Checks

+

-

Your profile

Languages

Preferences

Notifications

Account

Profile

Licenses

Audit log

API access

Watched projects

Automatically watch projects on contribution

Whenever you translate a string in a project, you will start watching it.

Watched projects

Search...

Available:

Web

lateOrg

Chosen:

Web

lateOrg

You can receive notifications for watched projects and they are shown on the dashboard by default.

Add all projects you want to translate to see them as watched projects on the dashboard.

Save

Notification settings

Other projects

Watched projects

Managed projects

Component wide notifications

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

Repository failure

Do not notify

Repository operation

Do not notify

Component locking

Do not notify

Changed license

Do not notify

Parse error

Do not notify

Comment on own translation

Instant notification

Mentioned in comment

Instant notification

New language

Do not notify

New translation component

Do not notify

New announcement

Instant notification

New alert

Do not notify

Translation notifications

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

New string

Do not notify

New contributor

Do not notify

New suggestion

Do not notify

New comment

Do not notify

Changed string

Do not notify

Translated string

Do not notify

Approved string

Do not notify

Pending suggestions

Do not notify

Unfinished strings

Do not notify

Save

Powered by Weblate 4.12

About Weblate

Legal

Contact

Documentation

Donate to Weblate

1.2. Registro y perfil de usuario

7

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

testuser

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

Full name

Weblate Test

E-mail

weblate@example.org

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:

- Publicly visible
- Visible solo a un grupo determinado de traductores

Ver también:

Control de acceso, Flujos de trabajo de traducción

1.3.1 Proyectos de traducción

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

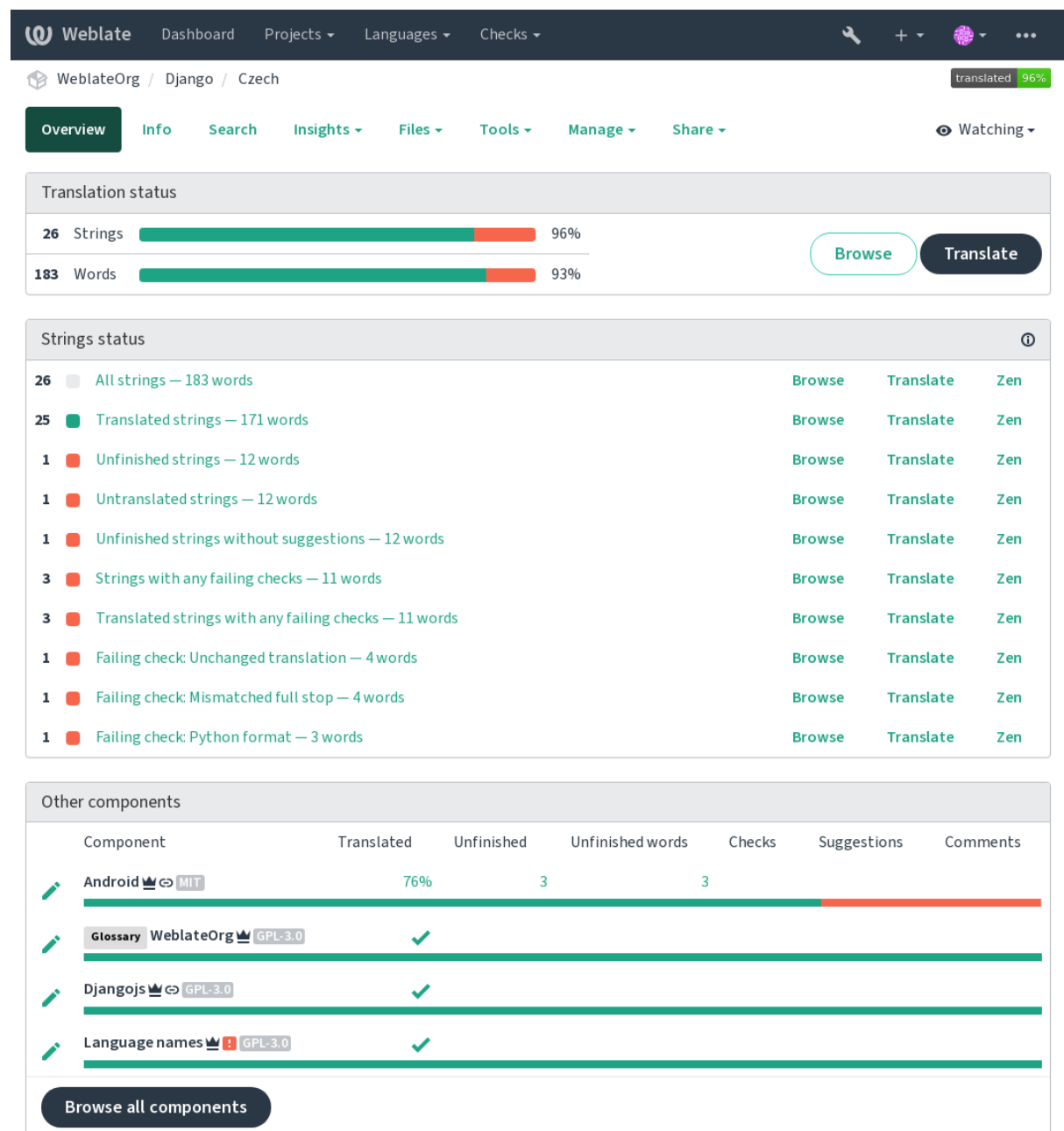
Component	Translated	Unfinished	Unfinished words	Checks	Suggestions	Comments
Android	79%	30	30	3		
Language names	95%	4	5	5		
Glossary WeblateOrg	100%	0	0	0		

[Add new translation component](#)

Powered by Weblate 4.12 [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 *Untranslated strings* or *Unfinished strings*. If the whole project is translated, without error, *All strings* is still available. Alternatively you can use the search field to find a specific string or term.



1.3.3 Sugerencias

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

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

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

1.3.4 Comentarios

Three types of comments can be posted: for translations, source strings, or to report source string bugs when this functionality is turned on using [Activar revisiones de origen](#). Choose the one suitable to topic you want to discuss. Source string comments are in any event good for providing feedback on the original string, for example that it should be rephrased or to ask questions about it.

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

Ver también:

report-source, [Source strings reviews](#), [Activar revisiones de origen](#)

1.3.5 Variantes

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

Ver también:

variants, [Variantes](#)

1.3.6 Etiquetas

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

Following labels are used by Weblate:

Traducida automáticamente String was translated using [Traducción automática](#).

La cadena de origen necesita revisión String was marked for review using [Source strings reviews](#).

Ver también:

labels

1.3.7 Traducción

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

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

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

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

Plurales

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

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

Ver también:

[Fórmula de plurales](#)

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

[WeblateOrg](#) / [Django](#) / [Czech](#) / [Translate](#)
translated 96%

[1 / 1](#)

[Custom search](#)

[Zen](#)

[Position and priority](#)

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 and continue

Save and stay

Suggest

Skip

[Nearby strings](#) 20
 [Comments](#)
[Automatic suggestions](#)
[Other languages](#) 3

[History](#)

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Glossary

English

Czech

No related strings found in the glossary.

[+ Add term to glossary](#)

String information

Screenshot context

No screenshot currently associated.

[+ Add screenshot](#)

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

6 seconds ago

Source string age

6 seconds ago

Translation file

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

Powered by Weblate 4.12

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

Atajos del 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+↑ or Alt+↑ or Cmd+↑	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	Submit current form; this is same as pressing <i>Save and continue</i> while editing translation.
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> checkbox.

Visual keyboard

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

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

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

17

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

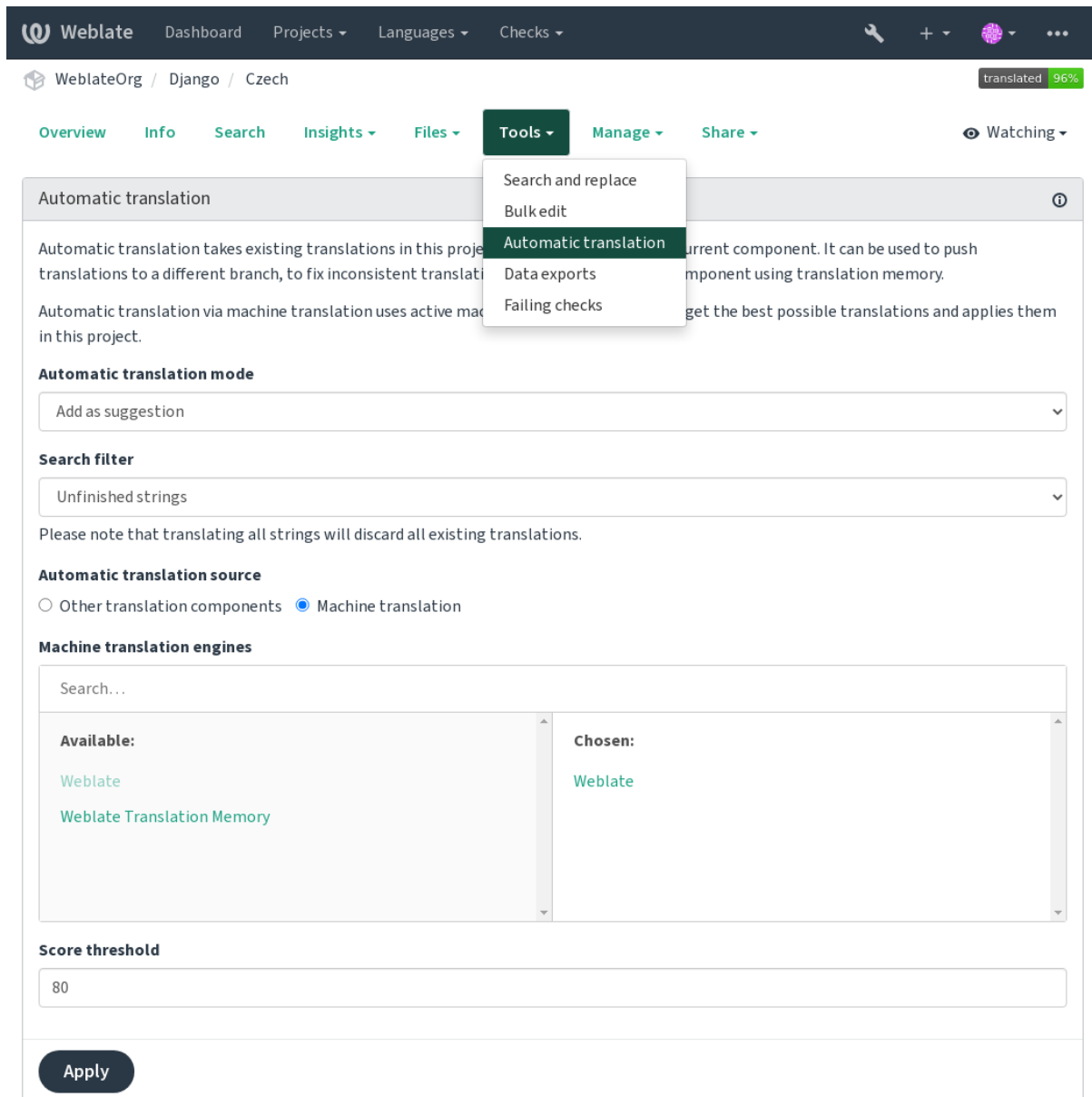
Based on configuration and your translated language, Weblate provides suggestions from several machine translation tools and [Memoria de traducción](#). All machine translations are available in a single tab of each translation page.

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 4.12 [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).

The automatically translated strings are labelled by *Automatically translated*.

Ver también:

Mantener iguales las traducciones entre los componentes

1.3.10 Rate limiting

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

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

1.3.11 Buscar y reemplazar

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

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

1.3.12 Edición en masa

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

- Changing string state (for example to approve all unreviewed strings).
- Adjust translation flags (see *Personalizar el comportamiento mediante indicadores*)
- Adjust string labels (see labels)

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

Ver también:

Bulk edit add-on

1.3.13 Matrix View

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

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

1.3.14 Zen Mode

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

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

1.4 Descargar y cargar traducciones

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

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

1.4.1 Descargar traducciones

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

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

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

- PO de gettext
- XLIFF con extensiones gettext
- XLIFF 1.1
- TermBase eXchange
- Translation Memory eXchange
- gettext MO (only available when translation is using gettext PO)
- CSV
- Open XML de Excel
- JSON (only available for monolingual translations)
- Android String Resource (only available for monolingual translations)
- iOS strings (only available for monolingual translations)

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

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / Django / Czech

translated 96%

Overview
Info
Search
Insights
Files
Tools
Manage
Share
Watching

Quick downloads

Download translation
Customize download
Upload translation

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

Customize download

All strings

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

Download

Powered by Weblate 4.12 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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. At the top, there's a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the breadcrumb 'WeblateOrg / Django / Czech' is visible, along with a 'translated 96%' indicator. The main navigation bar includes 'Overview', 'Info', 'Search', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. The 'Files' menu is open, showing 'Download translation', 'Customize download', and 'Upload translation'. The 'Upload' form is displayed, with a message: 'The uploaded file will be merged with the current file. Use the dropdown menu if you want to overwrite already translated strings.' The form includes a 'File' section with a 'Choose File' button and 'No file chosen' text. Below this is the 'File upload mode' section with four radio button options: 'Add as translation', 'Add as suggestion', 'Add as translation needing edit', and 'Replace existing translation file'. The 'Processing of strings needing edit' section has a dropdown menu set to 'Do not import'. The 'Conflict handling' section has a dropdown menu set to 'Update translated strings' with a note: 'Whether to overwrite existing translations if the string is already translated.' The 'Author name' field contains 'Weblate Test' and the 'Author e-mail' field contains 'weblate@example.org'. An 'Upload' button is at the bottom of the form.

Powered by Weblate 4.12 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, Descargar y cargar traducciones

Métodos de importación

Estas son las elecciones presentadas al cargar archivos de traducción:

Añadir como traducción (translate**)** Imported strings are added as translations to existing strings. This is the most common usecase, and the default behavior.

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

Añadir como sugerencia (suggest**)** Imported strings are added as suggestions, do this when you want to have your uploaded strings reviewed.

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

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

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

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

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

Solo determinados formatos de archivo admiten esta opción.

Añadir cadenas nuevas (add**)** Adds new strings to the translation. It skips the one which already exist.

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

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

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

Ver también:

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

Conflicts handling

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

Strings needing edit

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

Overriding authorship

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

1.5 Glosario

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

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

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.

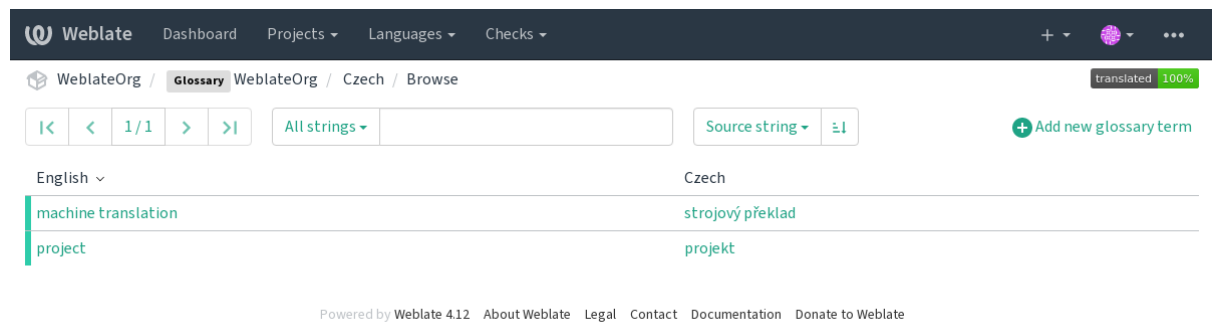
An empty glossary for a given project is automatically created with the project. Glossaries are shared among all components of the same project, and optionally with other projects using *Compartir en proyectos* from the respective glossary component.

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

The screenshot displays the Weblate web interface for managing a glossary. At the top, the navigation bar includes 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main header shows 'WeblateOrg / Glossary WeblateOrg / Czech' with a 'translated 100%' indicator. The 'Overview' tab is active, showing a 'Translation status' section with progress bars for 'Strings' (2) and 'Words' (3), both at 100%. To the right are buttons for 'Add new glossary term', 'Browse', and 'Translate'. Below this is the 'Strings status' section, showing 'All strings — 3 words' and 'Translated strings — 3 words', each with 'Browse', 'Translate', and 'Zen' links. The 'Other components' section features a table with columns for Component, Translated, Unfinished, Unfinished words, Checks, Suggestions, and Comments. It lists 'Django' (96% translated, 1 unfinished word, 12 unfinished words, 3 checks) and 'Language names' (100% translated). A 'Browse all components' button is located at the bottom of this section.

Powered by Weblate 4.12 About Weblate Legal Contact Documentation Donate to Weblate

Puede explorar todos los términos del glosario:

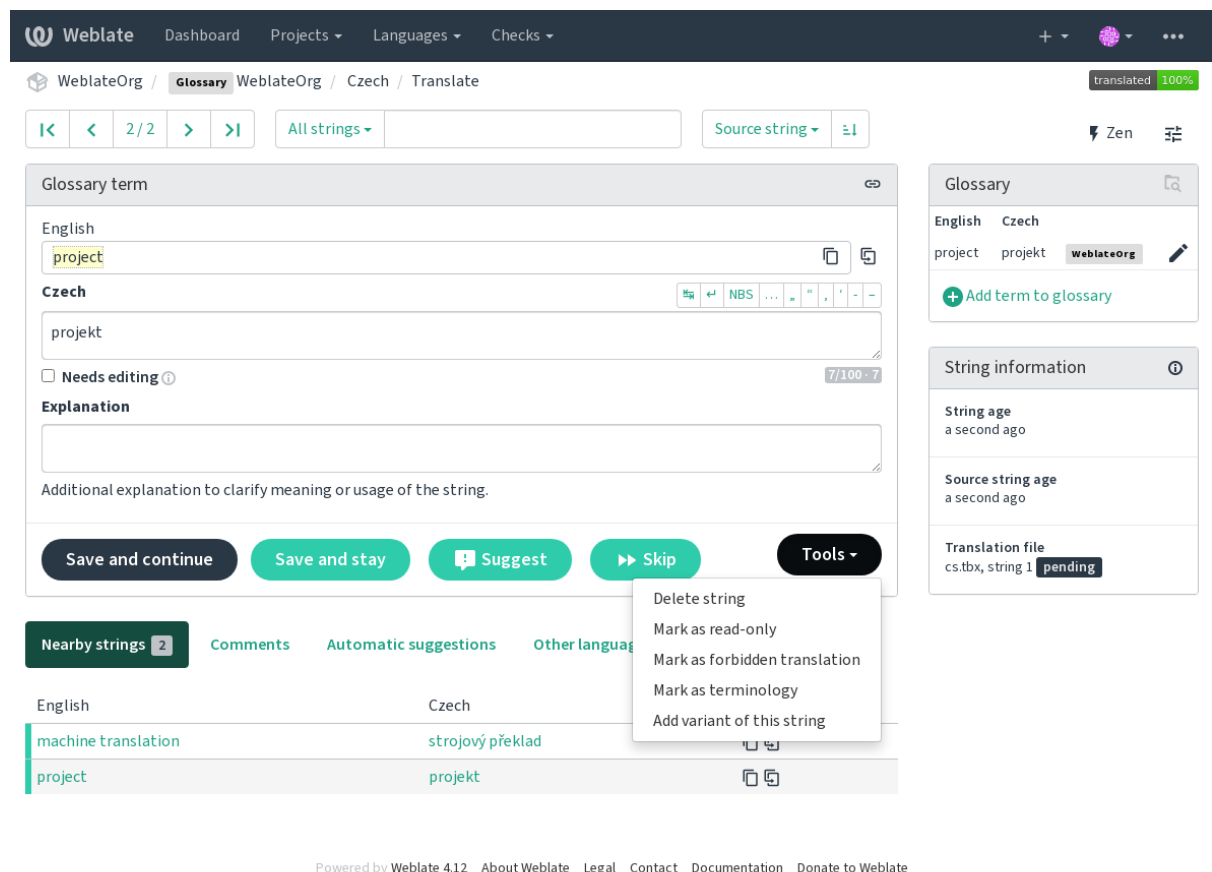


The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb trail reads 'WeblateOrg / Glossary WeblateOrg / Czech / Browse'. A 'translated 100%' badge is visible. The main area displays a table with two columns: 'English' and 'Czech'. The first row shows 'machine translation' and 'strojový překlad'. The second row shows 'project' and 'projekt'. Navigation controls at the top include 'All strings', 'Source string', and 'Add new glossary term'. At the bottom, a footer links to 'Powered by Weblate 4.12', 'About Weblate', 'Legal', 'Contact', 'Documentation', and '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.



This screenshot shows the 'Translate' view for a glossary term in Weblate. The main form has fields for 'English' (containing 'project') and 'Czech' (containing 'projekt'). Below these is a 'Needs editing' checkbox and an 'Explanation' field. At the bottom of the form are buttons: 'Save and continue', 'Save and stay', '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'. To the right, there's a 'Glossary' sidebar showing the current term and an 'Add term to glossary' button. Below that is a 'String information' sidebar showing 'String age', 'Source string age', and 'Translation file' (cs.tbx, string 1 pending). At the bottom, a 'Nearby strings' section shows a list of related terms. The footer is the same as the previous screenshot.

Untranslatable terms

Nuevo en la versión 4.5.

Flagging certain glossary term translations `read-only` by bulk-editing, typing in the flag, or by using *Tools* ↓ *Mark as 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 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:

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

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

translated 96%

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

<<

<

1 / 1

>

>>

Custom search

'%(count)s word'

Position

1

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 and continue

Save and stay

Suggest

Skip

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format 1

Following format strings are missing: %(count)s

Dismiss

For all languages

Missing plurals 2

Some plural forms are untranslated

Dismiss

For all languages

Glossary

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

9 seconds ago

Source string age

10 seconds ago

Translation file

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

pending

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

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

Alcance translated strings

Check class `weblate.checks.markup.BBCodeCheck`

Flag to ignore `ignore-bbcode`

BBCode represents simple markup, like for example highlighting important parts of a message in bold font, or italics. This check ensures they are also found in translation.

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

Palabras consecutivas duplicadas

Nuevo en la versión 4.1.

Summary El texto contiene la misma palabra dos veces seguidas:

Alcance translated strings

Check class `weblate.checks duplicate.DuplicateCheck`

Flag to ignore `ignore-duplicate`

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

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

No se ajusta al glosario

Nuevo en la versión 4.5.

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

Alcance translated strings

Check class `weblate.checks.glossary.GlossaryCheck`

Indicador que activar `check-glossary`

Flag to ignore `ignore-check-glossary`

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

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

Ver también:

Glosario, Personalizar el comportamiento mediante indicadores, Indicadores de traducción

Espacio duplicado

Summary La traducción contiene un espacio doble

Alcance translated strings

Check class `weblate.checks.chars.DoubleSpaceCheck`

Flag to ignore `ignore-double-space`

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

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

Cadenas formateadas

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 ▾

Zen

WeblateOrg / Django / Czech / Translate

translated 96%

1 / 1

Custom search ▾

'%(count)s word'

Position and priority ▾

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 and continue

Save and stay

Suggest

Skip

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

None

String updated in the repository

WeblateOrg / Django — Czech

English

Singular

%(count)s word

Plural

%(count)s words

Czech Translated

One

%(count)s slovo

Few

%(count)s slova

Other

%(count)s slov

6 seconds ago

Browse all component changes

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

6 seconds ago

Source string age

6 seconds ago

Translation file

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

Cadena de interpolación de AngularJS

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

Alcance translated strings

Check class `weblate.checks.angularjs.AngularJSInterpolationCheck`

Indicador que activar `angularjs-format`

Flag to ignore `ignore-angularjs-format`

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

Ver también:

Cadenas formateadas, [AngularJS text interpolation](#)

Formato C

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

Alcance translated strings

Check class `weblate.checks.format.CFormatCheck`

Indicador que activar `c-format`

Flag to ignore `ignore-c-format`

Simple format string example Hay `%d` manzanas

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

Ver también:

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

Formato C#

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

Alcance translated strings

Check class `weblate.checks.format.CSharpFormatCheck`

Indicador que activar `c-sharp-format`

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

Position format string example Hay `{0}` manzanas

Ver también:

Cadenas formateadas, [C# String Format](#)

Literales de plantilla ECMAScript

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

Alcance translated strings

Check class `weblate.checks.format.ESTemplateLiteralsCheck`

Indicador que activar `es-format`

Flag to ignore `ignore-es-format`

Interpolation example Hay `${number}` manzanas

Ver también:

Cadenas formateadas, *Template literals*

Interpolación con i18next

Nuevo en la versión 4.0.

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

Alcance translated strings

Check class `weblate.checks.format.I18NextInterpolationCheck`

Indicador que activar `i18next-interpolation`

Flag to ignore `ignore-i18next-interpolation`

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

Nesting example There are `$t(number)` apples

Ver también:

Cadenas formateadas, *i18next interpolation*

MessageFormat de ICU

Nuevo en la versión 4.9.

Summary Syntax errors and/or placeholder mismatches in ICU MessageFormat strings.

Alcance translated strings

Check class `weblate.checks.icu.ICUMessageFormatCheck`

Indicador que activar `icu-message-format`

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

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

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

xml	Enable support for simple XML tags. By default, XML tags are parsed loosely. Stray < characters are ignored if they are not reasonably part of a tag.
strict-xml	Enable support for strict XML tags. All < characters must be escaped if they are not part of a tag.
-highlight	Disable highlighting placeholders in the editor.
-require_other	Disable requiring sub-messages to have an other selector.
-submessage_selector	Skip checking that sub-message selectors match the source.
-types	Skip checking that placeholder types match the source.
-extra	Skip checking that no placeholders are present that were not present in the source string.
-missing	Skip checking that no placeholders are missing that were present in the source string.

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

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

Ver también:

[Sintaxis MessageFormat de ICU](#), [Cadenas formateadas](#), [ICU: Formatting Messages](#), [Format.JS: Message Syntax](#)

Formato Java

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

Alcance translated strings

Check class `weblate.checks.format.JavaFormatCheck`

Indicador que activar `java-format`

Flag to ignore `ignore-java-format`

Simple format string example Hay %d manzanas

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

Ver también:

[Cadenas formateadas](#), [Java Format Strings](#)

MessageFormat de Java

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

Alcance translated strings

Check class `weblate.checks.format.JavaMessageFormatCheck`

Flag to enable unconditionally `java-messageformat`

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

Flag to ignore `ignore-java-messageformat`

Position format string example Hay {0} manzanas

Ver también:

[Cadenas formateadas](#), [Java MessageFormat](#)

Formato JavaScript

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

Alcance translated strings

Check class `weblate.checks.format.JavaScriptFormatCheck`

Indicador que activar `javascript-format`

Flag to ignore `ignore-javascript-format`

Simple format string example Hay %d manzanas

Ver también:

Cadenas formateadas, *JavaScript formatting strings*

Formato Lua

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

Alcance translated strings

Check class `weblate.checks.format.LuaFormatCheck`

Indicador que activar `lua-format`

Flag to ignore `ignore-lua-format`

Simple format string example Hay %d manzanas

Ver también:

Cadenas formateadas, *Lua formatting strings*

Formato de Object Pascal

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

Alcance translated strings

Check class `weblate.checks.format.ObjectPascalFormatCheck`

Indicador que activar `object-pascal-format`

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

Simple format string example Hay %d manzanas

Ver también:

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

Sustitutorios con signo de porcentaje

Nuevo en la versión 4.0.

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

Alcance translated strings

Check class `weblate.checks.format.PercentPlaceholdersCheck`

Indicador que activar `percent-placeholders`

Flag to ignore `ignore-percent-placeholders`

Simple format string example There are %number% apples

Ver también:

[Cadenas formateadas](#),

Formato Perl

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

Alcance translated strings

Check class `weblate.checks.format.PperlFormatCheck`

Indicador que activar `perl-format`

Flag to ignore `ignore-perl-format`

Simple format string example Hay %d manzanas

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

Ver también:

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

Formato PHP

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

Alcance translated strings

Check class `weblate.checks.format.PHPFormatCheck`

Indicador que activar `php-format`

Flag to ignore `ignore-php-format`

Simple format string example Hay %d manzanas

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

Ver también:

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

Formato de llaves de Python

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

Alcance translated strings

Check class `weblate.checks.format.PythonBraceFormatCheck`

Indicador que activar `python-brace-format`

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

Simple format string There are {} apples

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

Ver también:

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

Formato Python

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

Alcance translated strings

Check class `weblate.checks.format.PythonFormatCheck`

Indicador que activar `python-format`

Flag to ignore `ignore-python-format`

Simple format string Hay %d manzanas

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

Ver también:

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

Formato Qt

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

Alcance translated strings

Check class `weblate.checks.qt.QtFormatCheck`

Indicador que activar `qt-format`

Flag to ignore `ignore-qt-format`

Position format string example There are %1 apples

Ver también:

Cadenas formateadas, [Qt QString::arg\(\)](#)

Formato de plurales de Qt

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

Alcance translated strings

Check class `weblate.checks.qt.QtPluralCheck`

Indicador que activar `qt-plural-format`

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

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

Ver también:

Cadenas formateadas, [Qt i18n guide](#)

Formato Ruby

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

Alcance translated strings

Check class `weblate.checks.ruby.RubyFormatCheck`

Indicador que activar `ruby-format`

Flag to ignore `ignore-ruby-format`

Simple format string example Hay %d manzanas

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

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

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

Ver también:

Cadenas formateadas, *Ruby Kernel#sprintf*

Formato Scheme

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

Alcance translated strings

Check class `weblate.checks.format.SchemeFormatCheck`

Indicador que activar `scheme-format`

Flag to ignore `ignore-scheme-format`

Simple format string example Hay ~d manzanas

Ver también:

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

Formato de Vue I18n

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

Alcance translated strings

Check class `weblate.checks.format.VueFormattingCheck`

Indicador que activar `vue-format`

Flag to ignore `ignore-vue-format`

Named formatting There are {count} apples

Rails i18n formatting There are %{count} apples

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

Ver también:

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

Se había traducido

Summary Esta cadena se ha traducido en el pasado

Alcance all strings

Check class `weblate.checks.consistency.TranslatedCheck`

Flag to ignore `ignore-translated`

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

Incoherente

Summary This string has more than one translation in this project or is untranslated in some components.

Alcance all strings

Check class `weblate.checks.consistency.ConsistencyCheck`

Flag to ignore `ignore-inconsistent`

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

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

This check applies to all components in a project that have *Permitir propagación de traducciones* turned on.

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

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

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

Ver también:

Mantener iguales las traducciones entre los componentes

Kashida utilizado

Nuevo en la versión 3.5.

Summary Las letras decorativas kashida no deben usarse

Alcance translated strings

Check class `weblate.checks.chars.KashidaCheck`

Flag to ignore `ignore-kashida`

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

Ver también:

[Kashida on Wikipedia](#)

Enlaces de Markdown

Nuevo en la versión 3.5.

Summary Los enlaces de descuento no coinciden con la fuente original

Alcance translated strings

Check class `weblate.checks.markup.MarkdownLinkCheck`

Indicador que activar `md-text`

Flag to ignore `ignore-md-link`

Markdown links do not match source.

Ver también:

[Enlaces de Markdown](#)

Referencias de Markdown

Nuevo en la versión 3.5.

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

Alcance translated strings

Check class `weblate.checks.markup.MarkdownRefLinkCheck`

Indicador que activar `md-text`

Flag to ignore `ignore-md-reflink`

Markdown link references do not match source.

Ver también:

[Markdown links](#)

Sintaxis de Markdown

Nuevo en la versión 3.5.

Summary La sintaxis de Markdown no coincide con la de origen

Alcance translated strings

Check class `weblate.checks.markup.MarkdownSyntaxCheck`

Indicador que activar `md-text`

Flag to ignore `ignore-md-syntax`

La sintaxis de Markdown no coincide con la de origen

Ver también:

[Markdown span elements](#)

Longitud máxima de la traducción

Summary La traducción no debe exceder la longitud indicada

Alcance translated strings

Check class `weblate.checks.chars.MaxLengthCheck`

Indicador que activar `max-length`

Flag to ignore `ignore-max-length`

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

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

Consejo: This check looks at number of chars, what might not be the best metric when using proportional fonts to render the text. The [Tamaño máximo de la traducción](#) check does check actual rendering of the text.

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

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

Tamaño máximo de la traducción

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

Alcance translated strings

Check class `weblate.checks.render.MaxSizeCheck`

Indicador que activar `max-size`

Flag to ignore `ignore-max-size`

Nuevo en la versión 3.7.

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

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

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

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

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

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

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

Ver también:

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

\n desiguales

Summary Number of \n in translation does not match source

Alcance translated strings

Check class `weblate.checks.chars.EscapedNewlineCountingCheck`

Flag to ignore `ignore-escaped-newline`

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

Summary El original o la traducción no acaba con dos puntos

Alcance translated strings

Check class `weblate.checks.chars.EndColonCheck`

Flag to ignore `ignore-end-colon`

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

Ver también:

[Colon on Wikipedia](#)

Puntos suspensivos desiguales

Summary El original o la traducción no acaba con puntos suspensivos

Alcance translated strings

Check class `weblate.checks.chars.EndEllipsisCheck`

Flag to ignore `ignore-end-ellipsis`

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

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

Ver también:

[Ellipsis on Wikipedia](#)

Signo de exclamación desigual

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

Alcance translated strings

Check class `weblate.checks.chars.EndExclamationCheck`

Flag to ignore `ignore-end-exclamation`

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

Ver también:

[Exclamation mark on Wikipedia](#)

Punto final desigual

Summary El original o la traducción no acaba con un punto

Alcance translated strings

Check class `weblate.checks.chars.EndStopCheck`

Flag to ignore `ignore-end-stop`

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

Ver también:

[Full stop on Wikipedia](#)

Signo de interrogación desigual

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

Alcance translated strings

Check class `weblate.checks.chars.EndQuestionCheck`

Flag to ignore `ignore-end-question`

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

Ver también:

[Question mark on Wikipedia](#)

Punto y coma desigual

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

Alcance translated strings

Check class `weblate.checks.chars.EndSemicolonCheck`

Flag to ignore `ignore-end-semicolon`

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

Ver también:

[Semicolon on Wikipedia](#)

Salto de renglón desiguales

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

Alcance translated strings

Check class `weblate.checks.chars.NewLineCountCheck`

Flag to ignore `ignore-newline-count`

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

Summary Some plural forms are untranslated

Alcance translated strings

Check class `weblate.checks.consistency.PluralsCheck`

Flag to ignore `ignore-plurals`

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

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

Sustitutorios

Nuevo en la versión 3.9.

Summary A la traducción le faltan algunos sustitutorios

Alcance translated strings

Check class `weblate.checks.placeholders.PlaceholderCheck`

Indicador que activar `placeholders`

Flag to ignore `ignore-placeholders`

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

Nuevo en la versión 3.9.

Summary Falta un espacio indivisible antes del signo de puntuación doble

Alcance translated strings

Check class `weblate.checks.chars.PunctuationSpacingCheck`

Flag to ignore `ignore-punctuation-spacing`

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

Ver también:

[French and English spacing on Wikipedia](#)

Expresión regular

Nuevo en la versión 3.9.

Summary La traducción no coincide con la expresión regular

Alcance translated strings

Check class `weblate.checks.placeholders.RegexCheck`

Indicador que activar `regex`

Flag to ignore `ignore-regex`

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

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

Mismos plurales

Summary Algunas formas plurales se traducen de la misma manera

Alcance translated strings

Check class `weblate.checks.consistency.SamePluralsCheck`

Flag to ignore `ignore-same-plurals`

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

Salto de renglón al inicio

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

Alcance translated strings

Check class `weblate.checks.chars.BeginNewlineCheck`

Flag to ignore `ignore-begin-newline`

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

Ver también:

Salto de renglón al final

Espacios iniciales

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

Alcance translated strings

Check class `weblate.checks.chars.BeginSpaceCheck`

Flag to ignore `ignore-begin-space`

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

Salto de renglón al final

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

Alcance translated strings

Check class `weblate.checks.chars.EndNewlineCheck`

Flag to ignore `ignore-end-newline`

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

Ver también:

Salto de renglón al inicio

Espacio al final

Summary El original o la traducción no acaba con un espacio

Alcance translated strings

Check class `weblate.checks.chars.EndSpaceCheck`

Flag to ignore `ignore-end-space`

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

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

Traducción no modificada

Summary El origen y la traducción son idénticos

Alcance translated strings

Check class `weblate.checks.same.SameCheck`

Flag to ignore `ignore-same`

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

This check can help find strings mistakenly untranslated.

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

Ver también:

Configuración de componentes, Personalizar el comportamiento mediante indicadores

HTML inseguro

Nuevo en la versión 3.9.

Summary La traducción utiliza marcación HTML insegura

Alcance translated strings

Check class `weblate.checks.markup.SafeHTMLCheck`

Indicador que activar `safe-html`

Flag to ignore `ignore-safe-html`

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

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

Ver también:

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

URL

Nuevo en la versión 3.5.

Summary La traducción no contiene un URL

Alcance translated strings

Check class `weblate.checks.markup.URLCheck`

Indicador que activar `url`

Flag to ignore `ignore-url`

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

Marcación XML

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

Alcance translated strings

Check class `weblate.checks.markup.XMLTagsCheck`

Flag to ignore `ignore-xml-tags`

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

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

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

Sintaxis XML

Nuevo en la versión 2.8.

Summary La traducción no constituye XML válido

Alcance translated strings

Check class `weblate.checks.markup.XMLValidityCheck`

Flag to ignore `ignore-xml-invalid`

The XML markup is not valid.

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

Espacio de anchura cero

Summary La traducción contiene espacios de anchura cero adicionales

Alcance translated strings

Check class `weblate.checks.chars.ZeroWidthSpaceCheck`

Flag to ignore `ignore-zero-width-space`

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

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

Ver también:

[Zero width space on Wikipedia](#)

1.6.4 Source checks

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

Puntos suspensivos

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

Alcance cadenas de origen

Check class `weblate.checks.source.EllipsisCheck`

Flag to ignore `ignore-ellipsis`

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

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

Ver también:

[Ellipsis on Wikipedia](#)

Sintaxis MessageFormat de ICU

Nuevo en la versión 4.9.

Summary Syntax errors in ICU MessageFormat strings.

Alcance cadenas de origen

Check class `weblate.checks.icu.ICUSourceCheck`

Indicador que activar `icu-message-format`

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

Ver también:

MessageFormat de ICU

Largamente no traducida

Nuevo en la versión 4.1.

Summary Esta cadena no se tradujo por mucho tiempo

Alcance cadenas de origen

Check class `weblate.checks.source.LongUntranslatedCheck`

Flag to ignore `ignore-long-untranslated`

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

Varias comprobaciones fallidas

Summary Las traducciones en varios idiomas tienen comprobaciones fallidas

Alcance cadenas de origen

Check class `weblate.checks.source.MultipleFailingCheck`

Flag to ignore `ignore-multiple-failures`

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

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

Varias variables sin nombre

Nuevo en la versión 4.1.

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

Alcance cadenas de origen

Check class `weblate.checks.format.MultipleUnnamedFormatsCheck`

Flag to ignore `ignore-unnamed-format`

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

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

No pluralizada

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

Alcance cadenas de origen

Check class `weblate.checks.source.OptionalPluralCheck`

Flag to ignore `ignore-optional-plural`

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

For example with Gettext in Python it could be:

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

1.7 Búsquedas

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, target, and context strings.

Search

All strings ▾ Sort By ▾

Advanced query builder

Source strings ▾ Search for... ☐ Exact Strings with suggestions ▾

String changed after ▾ mm/dd/yyyy

Query examples

Review strings changed by other users	<code>changed:>=2022-03-20 AND NOT changed_by:testuser</code>	<input type="button" value="Add"/>
Translated strings	<code>state:>=translated</code>	<input type="button" value="Add"/>
Strings with comments	<code>has:comment</code>	<input type="button" value="Add"/>
Strings with any failing checks	<code>has:check</code>	<input type="button" value="Add"/>
Strings with suggestions from others	<code>has:suggestion AND NOT suggestion_author:testuser</code>	<input type="button" value="Add"/>
Approved strings with suggestions	<code>state:approved AND has:suggestion</code>	<input type="button" value="Add"/>
All untranslated strings added the past month	<code>added:>=2022-03-20 AND state:<=needs-editing</code>	<input type="button" value="Add"/>
Translated strings in a certain language	<code>is:translated AND language:cs</code>	<input type="button" value="Add"/>

Powered by Weblate 4.12 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [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 Source string description 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:TEXTO State search (approved, translated, needs-editing, empty, read-only), supports *Operadores de campo*.

pending:BOOLEANO String pending for flushing to VCS.

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

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

language:TEXTO String target language.

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

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

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

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

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

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

check:TEXTO String has failing check.

dismissed_check:TEXTO String has dismissed check.

comment:TEXTO Search in user comments.

resolved_comment:TEXT Search in resolved comments.

comment_author:TEXTO Filter by comment author.

suggestion:TEXTO Search in suggestions.

suggestion_author:TEXTO Filter by suggestion author.

explanation:TEXTO Buscar en las explicaciones.

label:TEXT Search in labels.

screenshot:TEXT Search in screenshots.

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:

Weblate
 Dashboard Projects Languages Checks

WebplateOrg / Django / Czech / Translate
 translated 96%

<

>

1 / 1

Custom search

'%(count)s word'

Zen

Position and priority

Translation

English

Singular

'%(count)s word'

Plural

'%(count)s words'

Czech, One

'%(count)s slovo'

Czech, Few

'%(count)s slova'

Czech, Other

'%(count)s slov'

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

☐ Needs editing

Save and continue

Save and stay

Suggest

Skip

Untranslated strings • state:empty

Unfinished strings • state:<translated

Translated strings • state:>=translated

Strings marked for edit • state:needs-editing

Strings with suggestions • has:suggestion

Strings with variants • has:variant

Strings with labels • has:label

Strings with context • has:context

Unfinished strings without suggestions • state:<translated AND NOT has:suggestion

Strings with comments • has:comment

Strings with any failing checks • has:check

Approved strings • state:approved

Strings waiting for review • state:translated

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

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

6 seconds ago

Source string age

6 seconds ago

Translation file

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

1.7.9 Ordenar los resultados

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

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The top navigation bar includes links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. The main content area shows a list of 'Nearby strings' with 16 items. A dropdown menu is open, showing sorting options: 'Position and priority', 'Position', 'Priority', 'Labels', 'Source string', 'Target string', 'String age', 'Number of words', 'Number of comments', 'Number of failing checks', and 'Key'. The selected string is '(...) instead of an ellipsis character (...)'. The interface also features a 'Comments' section with a 'New comment' form and a 'String information' sidebar on the right. The sidebar includes sections for 'Glossary', 'String information', 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

1.8 Flujos de trabajo de traducción

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

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

1.8.1 Acceso de traducción

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

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

1.8.2 Estados de traducción

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

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

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

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

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

This state is only available when reviews are enabled.

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

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

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

Ver también:

Translation types capabilities, Flujos de trabajo de traducción

1.8.3 Traducción directa

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

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

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

1.8.4 Revisión por pares

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

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

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

1.8.5 Revisores dedicados

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

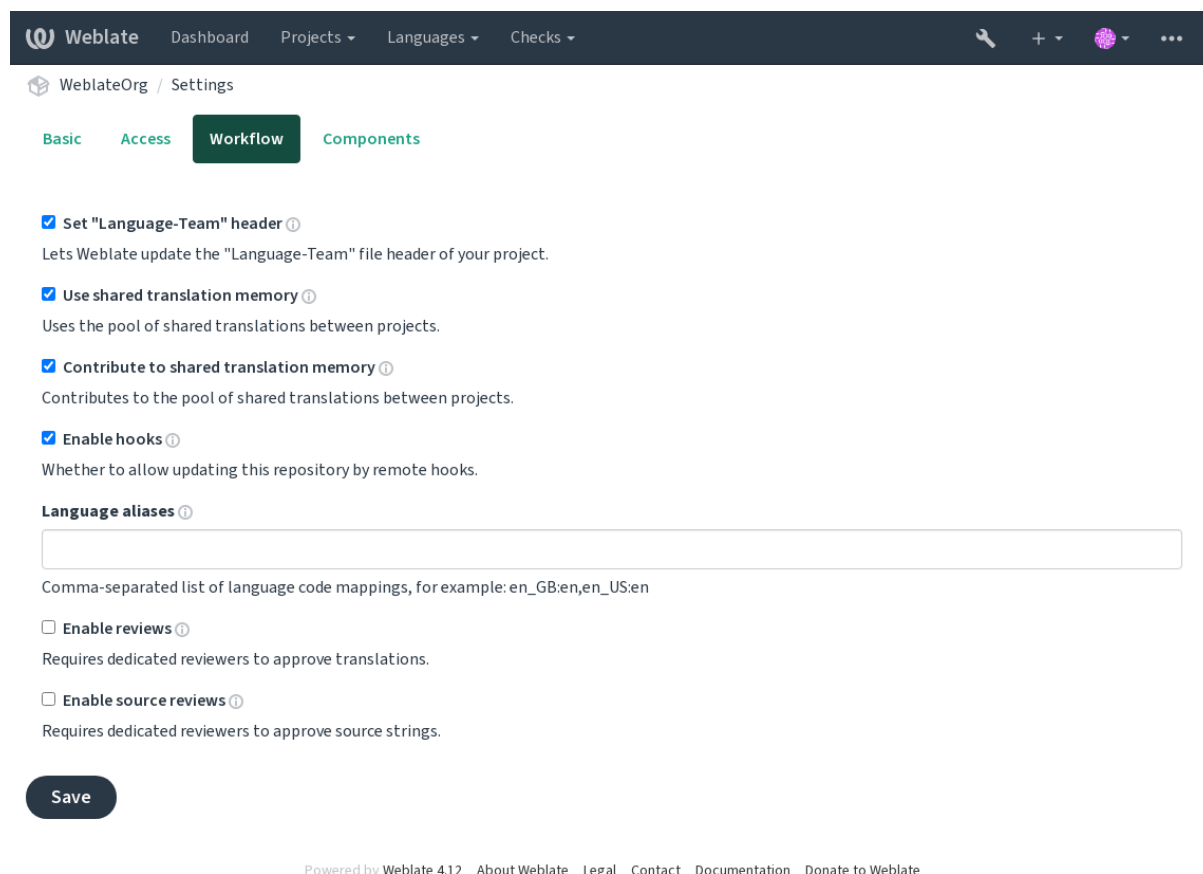
Al activar los revisores dedicados dividirá sus usuarios en dos grupos: uno será capaz de enviar traducciones, y el otro podrá revisarlas para garantizar que sean coherentes y su calidad sea buena.

- *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>Usuarios</i>	O bien, <i>traducir</i> con <i>control de acceso por proyecto</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):



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 breadcrumb trail: WeblateOrg / Settings. The main content area has a tabbed interface with 'Basic', 'Access', 'Workflow' (selected), and 'Components'. Under the 'Workflow' tab, there are several settings:

- ☒ **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** ⓘ
A text input field for a 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.

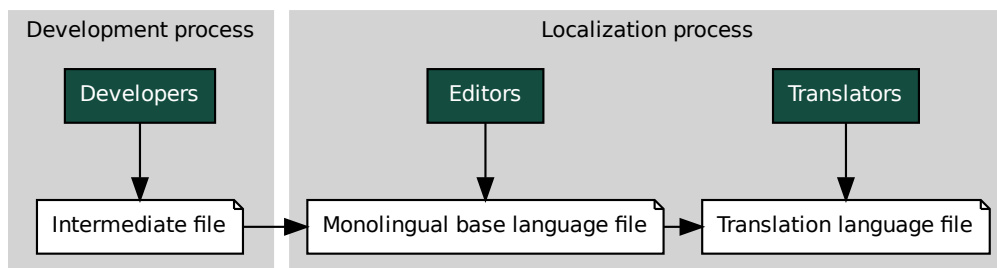
At the bottom left of the settings area is a dark 'Save' button. At the bottom of the page is a footer with the text: Powered by Weblate 4.12 About Weblate Legal Contact Documentation Donate to Weblate.

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

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

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

# Update weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

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

# 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

# Merge main branch:
git checkout main
git merge weblate-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 by `DEFAULT_COMMITTER_EMAIL` and `DEFAULT_COMMITTER_NAME`.

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

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

Ver también:

Configuración de componentes

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

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

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

1. Lock the affected component in Weblate.
2. Commit any pending changes and merge them into the upstream repository.
3. Disable receiving webhooks the *Configuración de proyectos*; this prevents Weblate from immediately seeing changes in the repository.
4. Do any needed changes in the repo (for example using `git mv`), push them to the upstream repository.
5. Change the *Configuración de componentes* to match the new setup; upon changing configuration, Weblate will fetch the updated repository and notice the changed locations while keeping existing strings.
6. Unlock the component and re-enable hooks in the project configuration.

1.9.2 Uso

How do I review the translations of others?

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

Ver también:

Flujos de trabajo de traducción

How do I provide feedback on a source string?

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

Ver también:

report-source, *Comentarios*

¿Cómo puedo utilizar traducciones existentes mientras traduzco?

- Todas las traducciones que se realicen dentro de Weblate pueden reutilizarse gracias a su memoria de traducción compartida.
- Es posible importar archivos de memoria de traducción existentes en Weblate.
- Use the import functionality to load compendium as translations, suggestions or translations needing review. This is the best approach for a one-time translation using a compendium or a similar translation database.
- You can set up *tmserver* with all databases you have and let Weblate use it. This is good when you want to use it several times during translation.
- Another option is to translate all related projects in a single Weblate instance, which will make it automatically pick up translations from other projects as well.

Ver también:

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:

Definiciones de idioma

Can Weblate highlight changes in a fuzzy string?

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

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

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

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

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

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

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

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

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

In case you want to do the update automatically, you can install add-on *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
```

(continúe en la próxima página)

(proviene de la página anterior)

```
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 [Procesamiento de códigos de idioma](#) for more details.

1.9.4 Funciones

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

Weblate currently does not have native support for anything other than [Git](#) (with extended support for [GitHub pull requests](#), [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 5646](#) to better indicate that they are really different languages instead previously wrongly used modifiers (for `@latin` variants) or country codes (for Chinese).

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

Nota: Weblate defaults to POSIX style language codes with underscore, see [Definiciones de idioma](#) for more details.

Ver también:

[Definiciones de idioma](#), [Estilo de código de idioma](#), [Añadir traducciones nuevas](#)

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 70, 1}	Plura- les ^{Página 70, 2}	Descrip- tions ^{Página 70, 3}	Contex- to ^{Página 70, 4}	Ubica- ción ^{Página 70, 5}	Flags ^{Página 70, 8}	Estados adiciona- les ^{Página 70, 6}
<i>gettext de GNU</i>	bilingual	yes	yes	yes	yes	yes ⁹	needs edi- ting
<i>gettext mo- nolingüe</i>	mono	yes	yes	yes	yes	yes ^{Página 70, 9}	needs edi- ting
<i>XLIFF</i>	both	yes	yes	yes	yes	yes ¹⁰	needs editing, approved
<i>Propie- dades de Java</i>	both	no	yes	no	no	no	
<i>mi18n lang files</i>	mono	no	yes	no	no	no	
<i>Propie- dades GWT</i>	mono	yes	yes	no	no	no	
<i>Traduccio- nes para Joomla</i>	mono	no	yes	no	yes	no	
<i>.ts de Qt Linguist</i>	both	yes	yes	no	yes	yes ^{Página 70, 10}	needs edi- ting
<i>Recursos de cadenas de Android</i>	mono	yes	yes ⁷	no	no	yes ^{Página 70, 10}	
<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	

continué en la próxima página

Tabla 1 – proviene de la página anterior

Formato	Lingua- lity ^{Página 70, 1}	Plura- les ^{Página 70, 2}	Descrip- tions ^{Página 70, 3}	Contex- to ^{Página 70, 4}	Ubica- ción ^{Página 70, 5}	Flags ^{Página 70, 8}	Estados adiciona- les ^{Página 70, 6}
<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 WebExtension</i>	mono	yes	yes	no	no	no	
<i>.XML resource files</i>	mono	no	yes	no	no	yes ^{Página 70, 10}	
<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 ^{Página 70, 10}	
<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 metadatos de tiendas de aplicaciones</i>	mono	no	no	no	no	no	
<i>Archivos de subtítulos</i>	mono	no	no	no	yes	no	
<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	

continué en la próxima página

Tabla 1 – proviene de la página anterior

Formato	Lingua- lity ^{Página 70, 1}	Plura- les ^{Página 70, 2}	Descrip- tions ^{Página 70, 3}	Contex- to ^{Página 70, 4}	Ubica- ción ^{Página 70, 5}	Flags ^{Página 70, 8}	Estados adiciona- les ^{Página 70, 6}
<i>TermBase eXchange format</i>	bilingual	no	yes	no	no	yes ^{Página 70, 10}	
<i>Archivos de texto</i>	mono	no	no	no	no	no	
<i>Stringsdict format</i>	mono	yes	yes	no	no	no	
<i>Fluent for- mat</i>	mono	no ¹²	yes	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 s  twer 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 "Pond  l  "

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

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

¹ Vea *Formatos biling  es y monoling  es*

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

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

⁴ El contexto se utiliza para discernir cadenas id  nticas que se utilizan en   mbitos distintos (por ejemplo, *Sun* puede emplearse como nombre abreviado del d  a «Sunday» o como el nombre de la estrella m  s cercana a nosotros).

⁵ La ubicaci  n en c  digo fuente de una cadena puede ayudar a los traductores avezados a entender de qu   manera una determinada cadena se utilizar  .

⁸ See *Personalizar el comportamiento mediante indicadores*

⁶ Additional states supported by the file format in addition to «Untranslated» and «Translated».

⁹ The gettext type comments are used as flags.

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

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

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

¹² Plurals are handled in the syntax of the strings and not exposed as plurals in Weblate.

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)*

gettext monolingüe

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

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

Estados de traducción

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

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

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

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

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

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

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

Ver también:

Revisores dedicados

Whitespace and newlines in XLIFF

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

Por ejemplo:

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

Especificar indicadores de traducción

You can specify additional translation flags (see *Personalizar el comportamiento mediante indicadores*) by using the `weblate-flags` attribute. Weblate also understands `maxwidth` and `font` attributes from the XLIFF specification:

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

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

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	
Máscara de archivos	<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 Propiedades de Java

El formato nativo de Java para las traducciones.

Java properties are usually used as monolingual translations.

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

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

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>Propiedades de Java (ISO-8859-1)</i>

Ver también:

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

1.10.7 mi18n lang files

Nuevo en la versión 4.7.

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

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

Ver también:

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

1.10.8 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, updating-target-files, *Formatea el archivo de propiedades de Java*, *Limpieza de archivos de traducción*

1.10.9 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 *Propiedades de Java* instead.

Ver también:

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

1.10.10 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*

1.10.11 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:

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

1.10.12 .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*

1.10.13 Recursos de cadenas de Android

Formato de archivo específico de Android para traducir aplicaciones.

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

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>

1.10.14 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:

Stringsdict format, Apple «strings files» documentation, Mac OSX strings

1.10.15 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](#)

1.10.16 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	<code>langs/translation-*.json</code>
Archivo de base monolingüe	<code>langs/translation-en.json</code>
Plantilla para traducciones nuevas	<i>Empty</i>
Formato de archivo	<i>JSON nested structure file</i>

Ver también:

[JSON](#), [updating-target-files](#), [Personalizar la salida de JSON](#), [Limpieza de archivos de traducción](#),

1.10.17 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.

The v4 variant uses different approach for storing plurals and is currently not supported.

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](#), [updating-target-files](#), [Personalizar la salida de JSON](#), [Limpieza de archivos de traducción](#)

1.10.18 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](#), [updating-target-files](#), [Personalizar la salida de JSON](#), [Limpieza de archivos de traducción](#),

1.10.19 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](#), [updating-target-files](#), [Personalizar la salida de JSON](#), [Limpieza de archivos de traducción](#)

1.10.20 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": "",
    "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](#)

1.10.21 .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 \(.resx\)](#), [updating-target-files](#), [Limpieza de archivos de traducción](#)

1.10.22 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	<code>locale/*.csv</code>
Archivo de base monolingüe	<i>Empty</i>
Plantilla para traducciones nuevas	<code>locale/en.csv</code>
Formato de archivo	<i>Archivo CSV</i>

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

Ver también:

[CSV](#)

1.10.23 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](#)

1.10.24 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: ""
    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](#)

1.10.25 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](#)

1.10.26 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](#)

1.10.27 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"
```

(continué en la próxima página)

(proviene de la página anterior)

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](#)

1.10.28 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](#).

1.10.29 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](#)

1.10.30 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.

1.10.31 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](#)

1.10.32 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.

Hay tres variedades de este formato:

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

Ver también:

[Simple Text Documents](#)

1.10.33 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](#)

1.10.34 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.

1.10.35 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	tbx/*.*tbx
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](#)

1.10.36 Stringsdict format

Nuevo en la versión 4.8.

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

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

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

Ver también:

[Cadenas de iOS de Apple](#), [Stringsdict File Format](#)

1.10.37 Fluent format

Nuevo en la versión 4.8.

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

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

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

Ver también:

[Project Fluent website](#)

1.10.38 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 pull requests](#), [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 web interface. At the top is a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a sub-header 'Manage / SSH keys'. A horizontal menu contains links: 'Weblate status', 'Backups', 'Translation memory', 'Performance report', 'SSH keys' (highlighted), 'Alerts', 'Repositories', 'Users', and 'Appearance'. Below the menu, there are two sections. The first section, 'Public SSH key', has a title bar with an info icon. It contains a text box with the public key: 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADFDJvy5MAjwiRJRfYgV0CNSjKq+bNFDERlctPqNSkPY6jAUrm8aLviU5d9ngeykeqc0rT08OBAmuKqt9Gn+N6soAt2xEqb2i3xs0 Weblate'. Below the key is a 'Download private key' button. The second section, 'Known host keys', has a title bar with an info icon. It contains a table with three columns: 'Hostname', 'Key type', and 'Fingerprint'. The table lists three entries for 'github.com'. Below the table is an 'Add host key' section with a title bar and an info icon. It contains a text box with the instruction: '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.' Below this is a form with 'Hostname' and 'Port' labels and input fields. At the bottom of the form is a 'Submit' button. At the very bottom of the page, there is a footer with the text: 'Powered by Weblate 4.12 About Weblate Legal Contact Documentation Donate to Weblate'.

Public SSH key

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

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADFDJvy5MAjwiRJRfYgV0CNSjKq+bNFDERlctPqNSkPY6jAUrm8aLviU5d9ngeykeqc0rT08OBAmuKqt9Gn+N6soAt2xEqb2i3xs0
Weblate
```

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-ed25519	+DIY3wwV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-rsa	nThbg6kXUpJWGf7E1IGOCspRomTxdCARLviKw6E5SY8

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 4.12 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.

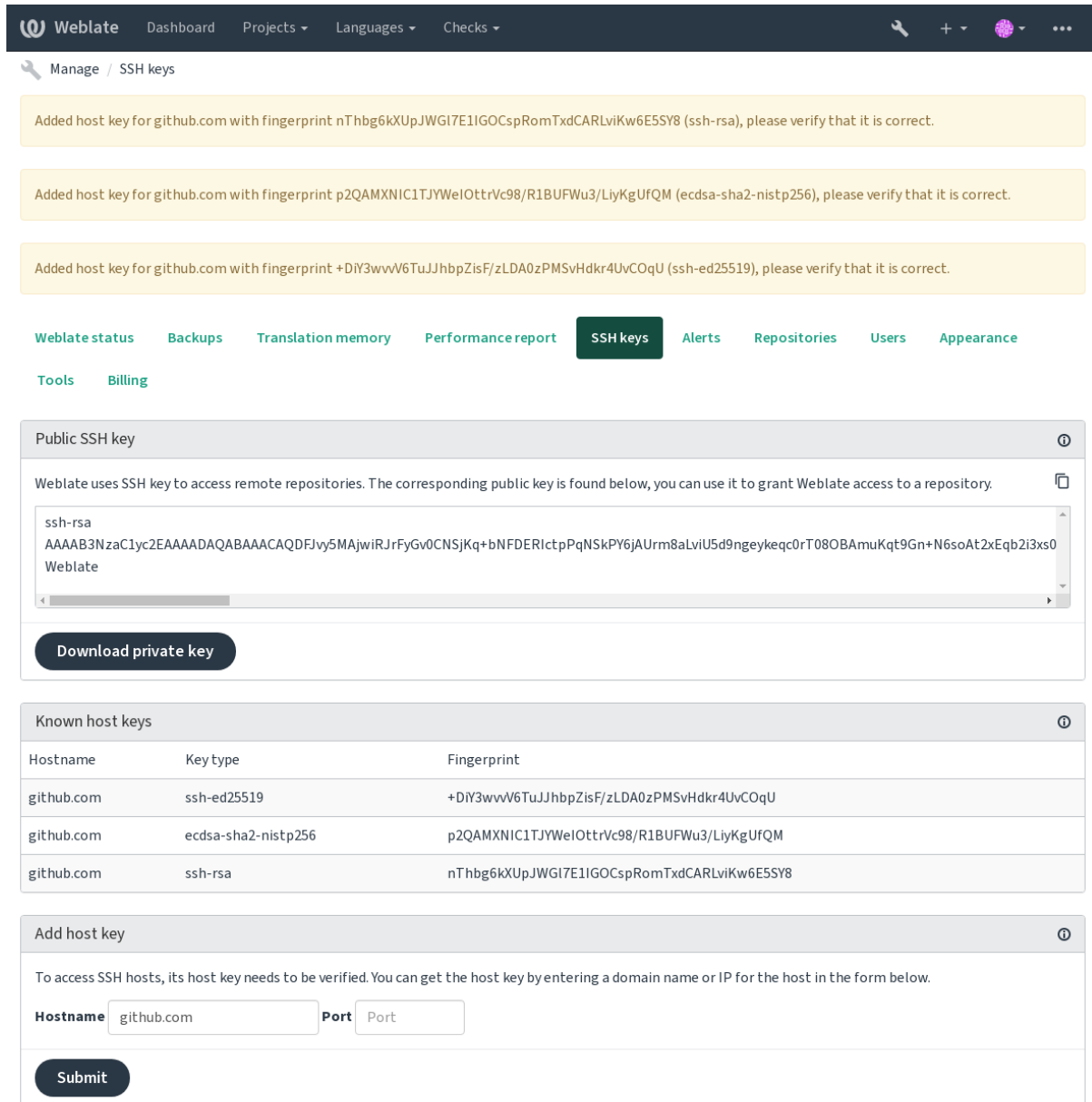
Consejo: Make a backup of the generated private Weblate SSH key.

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 displays the Weblate SSH keys management interface. At the top, there's a navigation bar with 'Weblate' logo and links to Dashboard, Projects, Languages, and Checks. Below this, a 'Manage / SSH keys' section shows three confirmation messages for added host keys for github.com with their respective fingerprints and key types (ssh-rsa, ecdsa-sha2-nistp256, and ssh-ed25519). A sidebar on the right contains links to Weblate status, Backups, Translation memory, Performance report, SSH keys (active), Alerts, Repositories, Users, Appearance, Tools, and Billing. The main content area is divided into three sections: 'Public SSH key' showing a key for Weblate access, 'Known host keys' displaying a table of host keys, and 'Add host key' providing a form to add a new host key.

Public SSH key

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

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDEJvy5MAjwiRjrFyGv0CNSjKq+bNFDERlctPqNSkPY6jAUrm8aLviU5d9ngeykeqc0rT08OBAmuKqt9Gn+N6soAt2xEqb2i3xs0
Weblate
```

[Download private key](#)

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-ed25519	+DiY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM
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](#)

Repositorios en GitHub

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

In case the [Rama a la que enviar](#) is not set, the project is forked and changes pushed through a fork. In case it is set, changes are pushed to the upstream repository and chosen branch.

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

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

Ver también:

[Accessing repositories from Hosted Weblate](#)

URL internos de Weblate

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

Advertencia: Removing main component also removes linked components.

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

Reasons to use this:

- Ahorra espacio en disco en el servidor, ya que el repositorio se almacena solo una vez.
- Acelera las actualizaciones, ya que se actualiza solo un repositorio.
- There is just single exported repository with Weblate translations (see [Git exporter](#)).
- Some add-ons can operate on multiple components sharing one repository, for example [Concentrar consignas de Git](#).

Repositorios HTTPS

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

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

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

Using proxy

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

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

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

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

Ver también:

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

1.11.2 Git

Consejo: Weblate needs Git 2.12 or newer.

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 pull requests

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 pull requests* creates pull requests. The latter is not needed for merely accessing Git repositories.

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

Ver también:

Enviar cambios efectuados en Weblate, GITHUB_USERNAME, GITHUB_TOKEN, GITHUB_CREDENTIALS

1.11.4 GitLab merge requests

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 merge requests* creates merge request.

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

Ver también:

Enviar cambios efectuados en Weblate, GITLAB_USERNAME, GITLAB_TOKEN, GITLAB_CREDENTIALS

1.11.5 Gitea pull requests

Nuevo en la versión 4.12.

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

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

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

Ver también:

Enviar cambios efectuados en Weblate, GITEA_USERNAME, GITEA_TOKEN, GITEA_CREDENTIALS

1.11.6 Pagure merge requests

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 merge requests](#) creates merge request.

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

Ver también:

Enviar cambios efectuados en Weblate, `PAGURE_USERNAME`, `PAGURE_TOKEN`, `PAGURE_CREDENTIALS`

1.11.7 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.

1.11.8 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.

1.11.9 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`

1.11.10 Archivos locales

1.11.11 Git

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

Nuevo en la versión 3.8.

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

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

1.12 API REST de Weblate

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.
- **page** – Returns given page of paginated results (use *next* and *previous* fields in response to automate the navigation).

Request Headers

- *Accept* – the response content type depends on *Accept* header

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

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 deleted successfully
- 400 Bad Request – when form parameters are missing
- 403 Forbidden – when access is denied
- 429 Too Many Requests – when throttling is in place

Authentication tokens

Distinto en la versión 4.10: Project scoped tokens were introduced in the 4.10 release.

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

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

Authentication examples

Example request:

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

Example response:

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

(continué en la próxima página)

(proviene de la página anterior)

```
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.

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

The status of rate limiting is reported in following headers:

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

Distinto en la versión 4.1: Added ratelimiting status headers.

Ver también:

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

1.12.2 API Entry Point

GET /api/

The API root entry point.

Example request:

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

Example response:

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

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

1.12.3 Usuarios

Nuevo en la versión 4.0.

GET /api/users/

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

Ver también:

Users object attributes are documented at [GET /api/users/\(str:username\)/](#).

POST /api/users/

Creates a new user.

Parámetros

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

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

Returns information about users.

Parámetros

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

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
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – date the user is created
- **groups** (*array*) – link to associated groups; see [GET /api/groups/\(int:id\)/](#)

Example JSON data:

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

(continué en la próxima página)

(proviene de la página anterior)

```

"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
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – date the user is created

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

Changes the user parameters.

Parámetros

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

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
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – date the user is created

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

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

Parámetros

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

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

Associate groups with a user.

Parámetros

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

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
- **defining_project** (*str*) – link to the defining project, used for *Gestionar el control de acceso por proyecto*; see [GET /api/projects/\(string:project\)/](#)

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)/*
- **componentlists** (*array*) – link to associated componentlist; see *GET /api/component-lists/(str:slug)/*
- **defining_project** (*str*) – link to the defining project, used for *Gestionar el control de acceso por proyecto*; see *GET /api/projects/(string:project)/*

Example JSON data:

```
{
  "name": "Guests",
  "defining_project": null,
  "project_selection": 3,
  "language_selection": 1,
  "url": "http://example.com/api/groups/1/",
  "roles": [
    "http://example.com/api/roles/1/",
    "http://example.com/api/roles/2/"
  ],
  "languages": [
    "http://example.com/api/languages/en/",
    "http://example.com/api/languages/cs/"
  ],
  "projects": [
    "http://example.com/api/projects/demo1/",
    "http://example.com/api/projects/demo/"
  ],
  "componentlist": "http://example.com/api/component-lists/new/",
  "components": [
    "http://example.com/api/components/demo/weblate/"
  ]
}
```

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

Cambia los parámetros del grupo.

Parámetros

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

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",
    "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
```

(continué en la próxima página)

(proviene de la página anterior)

```
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*.

Request JSON Object

- **object** – Component parameters, see *GET /api/components/(string:project)/(string:component)/*

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

JSON request to create a new component from Git:

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

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

JSON request to create a new component from another one:

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

{
  "file_format": "po",
  "filemask": "po/*.po",
  "name": "Weblate",
  "slug": "weblate",

```

(continué en la próxima página)

(proviene de la página anterior)

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

Ejemplo de respuesta en JSON

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

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

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

Returns paginated statistics for all languages within a project.

Nuevo en la versión 3.8.

Parámetros

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

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

Consejo: Use *POST* /api/projects/(string:project)/components/ to create new components.

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*) – Máscara de archivos
- **template** (*string*) – Archivo de base monolingüe
- **edit_template** (*string*) – Editar archivo de base
- **intermediate** (*string*) – Archivo de idioma intermediario
- **new_base** (*string*) – Plantilla para traducciones nuevas
- **file_format** (*string*) – Formato de archivo
- **license** (*string*) – Licencia de la traducción
- **agreement** (*string*) – Acuerdo de contribuidor
- **new_lang** (*string*) – Adición de traducciones nuevas
- **language_code_style** (*string*) – Estilo de código de idioma
- **source_language** (*object*) – objeto de idioma de origen; vea `GET /api/languages/(string:language)/`
- **push** (*string*) – URL de envío al repositorio
- **check_flags** (*string*) – Indicadores de traducción
- **priority** (*string*) – Prioridad
- **enforced_checks** (*string*) – Comprobaciones obligatorias
- **restricted** (*string*) – Acceso restringido
- **repoweb** (*string*) – Explorador del repositorio
- **report_source_bugs** (*string*) – Dirección para informar de errores en las cadenas de origen
- **merge_style** (*string*) – Estilo de fusión
- **commit_message** (*string*) – Commit, add, delete, merge, add-on, and merge request messages
- **add_message** (*string*) – Commit, add, delete, merge, add-on, and merge request messages
- **delete_message** (*string*) – Commit, add, delete, merge, add-on, and merge request messages
- **merge_message** (*string*) – Commit, add, delete, merge, add-on, and merge request messages
- **addon_message** (*string*) – Commit, add, delete, merge, add-on, and merge request messages
- **pull_message** (*string*) – Commit, add, delete, merge, add-on, and merge request messages
- **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",
    "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",
```

(continué en la próxima página)

(proviene de la página anterior)

```

        "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/file/`

Nuevo en la versión 4.9.

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

Parámetros

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

Query Parameters

- **format** (*string*) – The archive format to use; If not specified, defaults to zip; Supported formats: zip
- **q** (*string*) – Filter downloaded strings, see search.

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,
```

(continué en la próxima página)

(proviene de la página anterior)

```

"have_comment": 0,
"have_suggestion": 0,
"is_template": false,
"is_source": 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",
"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/"
    }
  }
}
```

(continué en la próxima página)

(proviene de la página anterior)

```

    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "slug": "weblate",
    "template": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
  },
  "failing_checks": 3,
  "failing_checks_percent": 75.0,
  "failing_checks_words": 11,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
    "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*) – Name of translation unit (used as key or context)
- **value** (*array*) – Source strings (use single string if not creating plural)

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*) – Automatic translation source - mt or others
- **component** (*string*) – Permita que el proyecto contribuya a la memoria de traducción compartida para obtener acceso a componentes adicionales.
- **engines** (*array*) – Motores de traducción automática

- **threshold** (*string*) – Umbral de puntuación

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

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

Nota: This API endpoint uses different logic for output than rest of API as it operates on whole file rather than on data. Set of accepted *format* parameter differs and without such parameter you get translation file as stored in VCS.

Query Parameters

- **format** – File format to use; if not specified no format conversion happens; supported file formats: po, mo, xliff, xliff11, 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 conflict** – 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, re-set, 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 - untranslated, 10 - needs editing, 20 - translated, 30 - approved, 100 - read only
- **fuzzy** (*boolean*) – si la unidad está marcada como pendiente de trabajo o revisión
- **translated** (*boolean*) – indica si la unidad está traducida
- **approved** (*boolean*) – indica si la traducción está aprobada
- **position** (*int*) – posición de la unidad en el archivo de traducción
- **has_suggestion** (*boolean*) – indica si la unidad tiene sugerencias
- **has_comment** (*boolean*) – indica si la unidad tiene comentarios
- **has_failing_check** (*boolean*) – indica si la unidad tiene comprobaciones fallidas
- **num_words** (*int*) – número de palabras de origen
- **priority** (*int*) – prioridad de traducción; 100 es la predeterminada
- **id** (*int*) – identificador de unidad
- **explanation** (*string*) – String explanation, available on source units, see [Información adicional sobre las cadenas de origen](#)

- **extra_flags** (*string*) – Additional string flags, available on source units, see [Personalizar el comportamiento mediante indicadores](#)
- **web_url** (*string*) – URL para editar la unidad
- **source_unit** (*string*) – Source unit link; see [GET /api/units/\(int:id\)/](#)
- **pending** (*boolean*) – whether the unit is pending for write
- **timestamp** (*timestamp*) – string age

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 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see [Revisores dedicados](#))
- **target** (*array*) – cadena de destino
- **explanation** (*string*) – String explanation, available on source units, see [Información adicional sobre las cadenas de origen](#)
- **extra_flags** (*string*) – Additional string flags, available on source units, see [Personalizar el comportamiento mediante indicadores](#)

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 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see [Revisores dedicados](#))
- **target** (*array*) – cadena de destino
- **explanation** (*string*) – String explanation, available on source units, see [Información adicional sobre las cadenas de origen](#)
- **extra_flags** (*string*) – Additional string flags, available on source units, see [Personalizar el comportamiento mediante indicadores](#)

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*) – Add-on ID

Response JSON Object

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

Ver también:

Complementos

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

Crea un complemento nuevo.

Parámetros

- **project_slug** (*string*) – «Slug» del proyecto
- **component_slug** (*string*) – «Slug» del componente

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*) – Add-on ID

Response JSON Object

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

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

Editar información completa relativa a un complemento.

Parámetros

- **id** (*int*) – Add-on ID

Response JSON Object

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

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

Eliminar el complemento.

Parámetros

- **id** (*int*) – Add-on ID

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 Metrics

GET `/api/metrics/`

Returns server metrics.

Response JSON Object

- **units** (*int*) – Number of units
- **units_translated** (*int*) – Number of translated units
- **users** (*int*) – Number of users
- **changes** (*int*) – Cantidad de cambios
- **projects** (*int*) – Number of projects
- **components** (*int*) – Número de componentes
- **translations** (*int*) – Number of translations
- **languages** (*int*) – Number of used languages
- **checks** (*int*) – Number of triggered quality checks
- **configuration_errors** (*int*) – Number of configuration errors
- **suggestions** (*int*) – Number of pending suggestions
- **celery_queues** (*object*) – Lengths of Celery queues, see *Tareas en segundo plano con Celery*
- **name** (*string*) – Configured server name

1.12.18 Actuadores de notificación

Los actuadores de notificación permiten a aplicaciones externas notificar a Weblate de que se ha actualizado un repositorio de control de versiones.

You can use repository endpoints for projects, components and translations to update individual repositories; see `POST /api/projects/(string:project)/repository/` for documentation.

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

Obsoleto desde la versión 2.6: Please use `POST /api/components/(string:project)/(string:component)/repository/` instead which works properly with authentication for ACL limited projects.

Triggers update of a component (pulling from VCS and scanning for translation changes).

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

Obsoleto desde la versión 2.6: Please use `POST /api/projects/(string:project)/repository/` instead which works properly with authentication for ACL limited projects.

Triggers update of all components in a project (pulling from VCS and scanning for translation changes).

POST `/hooks/github/`

Special hook for handling GitHub notifications and automatically updating matching components.

Nota: GitHub includes direct support for notifying Weblate: enable Weblate service hook in repository settings and set the URL to the URL of your Weblate installation.

Ver también:

Recibir cambios automáticamente de GitHub For instruction on setting up GitHub integration

<https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks>

Información genérica sobre los actuadores web de GitHub

ENABLE_HOOKS Para activar actuadores en todo Weblate

POST /hooks/gitlab/

Special hook for handling GitLab notifications and automatically updating matching components.

Ver también:

Automatically receiving changes from GitLab For instruction on setting up GitLab integration

<https://docs.gitlab.com/ee/user/project/integrations/webhooks.html> Información genérica sobre los actuadores web de GitLab

ENABLE_HOOKS Para activar actuadores en todo Weblate

POST /hooks/bitbucket/

Special hook for handling Bitbucket notifications and automatically updating matching components.

Ver también:

Recibir cambios automáticamente de Bitbucket For instruction on setting up Bitbucket integration

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/> Información genérica sobre los actuadores web de Bitbucket

ENABLE_HOOKS Para activar actuadores en todo Weblate

POST /hooks/pagure/

Nuevo en la versión 3.3.

Special hook for handling Pagure notifications and automatically updating matching components.

Ver también:

Recibir cambios automáticamente de Pagure For instruction on setting up Pagure integration

https://docs.pagure.org/pagure/usage/using_webhooks.html Información genérica sobre los actuadores web de Pagure

ENABLE_HOOKS Para activar actuadores en todo Weblate

POST /hooks/azure/

Nuevo en la versión 3.8.

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

1.12.19 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.

Example request:

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

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

[
  {
    "code": "cs",
    "failing": 0,
    "failing_percent": 0.0,
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "last_author": "Michal Čihař",
    "last_change": "2012-03-28T15:07:38+00:00",
    "name": "Czech",
    "total": 436,
    "total_words": 15271,
    "translated": 436,
    "translated_percent": 100.0,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/cs/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
```

(continué en la próxima página)

(proviene de la página anterior)

```

    },
    {
      "code": "nl",
      "failing": 21,
      "failing_percent": 4.8,
      "fuzzy": 11,
      "fuzzy_percent": 2.5,
      "last_author": null,
      "last_change": null,
      "name": "Dutch",
      "total": 436,
      "total_words": 15271,
      "translated": 319,
      "translated_percent": 73.2,
      "translated_words": 3201,
      "url": "http://hosted.weblate.org/engage/weblate/nl/",
      "url_translate": "http://hosted.weblate.org/projects/weblate/main/nl/"
    },
    {
      "code": "el",
      "failing": 11,
      "failing_percent": 2.5,
      "fuzzy": 21,
      "fuzzy_percent": 4.8,
      "last_author": null,
      "last_change": null,
      "name": "Greek",
      "total": 436,
      "total_words": 15271,
      "translated": 312,
      "translated_percent": 71.6,
      "translated_words": 3201,
      "url": "http://hosted.weblate.org/engage/weblate/el/",
      "url_translate": "http://hosted.weblate.org/projects/weblate/main/el/"
    }
  ]

```

1.12.20 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.

--method

Upload method to use, see *Métodos de importación*.

--fuzzy

Fuzzy (marked for edit) strings processing (*empty*, *process*, *approve*)

--author-name

Author name, to override currently authenticated user

--author-email

Author e-mail, to override currently authenticated user

Consejo: Puede obtener más información al invocar cada orden si añade `--help`; por ejemplo: `wlc ls --help`.

1.13.6 Archivos de configuración

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

Cargar un archivo de traducción:

```
$ wlc upload project/component/language --input /tmp/hello.po
```

También puede designar el proyecto en el que debe funcionar `wlc`:

```
$ cat .weblate
[weblate]
url = https://hosted.weblate.org/api/
translation = weblate/application

$ wlc show
branch: main
file_format: po
source_language: en
filemask: weblate/locale/*/LC_MESSAGES/django.po
git_export: https://hosted.weblate.org/git/weblate/application/
license: GPL-3.0+
license_url: https://spdx.org/licenses/GPL-3.0+
name: Application
new_base: weblate/locale/django.pot
project: weblate
repo: git://github.com/WeblateOrg/weblate.git
slug: application
template:
url: https://hosted.weblate.org/api/components/weblate/application/
vcs: git
web_url: https://hosted.weblate.org/projects/weblate/application/
```

Con esta configuración es sencillo consignar los cambios pendientes del proyecto actual:

```
$ wlc commit
```

1.14 API de Python de Weblate

1.14.1 Instalación

The Python API is shipped separately, you need to install the *Cliente de Weblate* (wlc) to have it.

```
pip install wlc
```

1.14.2 wlc

`WeblateException`

exception `wlc.WeblateException`

Base class for all exceptions.

`Weblate`

class `wlc.Weblate` (*key=""*, *url=None*, *config=None*)

Parámetros

- **key** (*str*) – User key
- **url** (*str*) – API server URL, if not specified default is used
- **config** (`wlc.config.WeblateConfig`) – Configuration object, overrides any other parameters.

Access class to the API, define API key and optionally API URL.

get (*path*)

Parámetros **path** (*str*) – Request path

Tipo del valor devuelto object

Performs a single API GET call.

post (*path*, ***kwargs*)

Parámetros **path** (*str*) – Request path

Tipo del valor devuelto object

Performs a single API GET call.

1.14.3 `wlc.config`

`WeblateConfig`

class `wlc.config.WeblateConfig` (*section='wlc'*)

Parámetros **section** (*str*) – Configuration section to use

Configuration file parser following XDG specification.

load (*path=None*)

Parámetros **path** (*str*) – Path from which to load configuration.

Loads configuration from a file, if none is specified, it loads from the *wlc* configuration file (`~/ .config/ wlc`) placed in your XDG configuration path (`/etc/xdg/wlc`).

1.14.4 `wlc.main`

`wlc.main.main` (*settings=None, stdout=None, args=None*)

Parámetros

- **settings** (*list*) – Settings to override as list of tuples
- **stdout** (*object*) – stdout file object for printing output, uses `sys.stdout` as default
- **args** (*list*) – Command-line arguments to process, uses `sys.args` as default

Main entry point for command-line interface.

`@wlc.main.register_command` (*command*)

Decorator to register *Command* class in main parser used by *main()*.

Command

class `wlc.main.Command` (*args, config, stdout=None*)

Clase principal para invocar órdenes.

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

Choosing Docker hub tag

You can use following tags on Docker hub, see <https://hub.docker.com/r/weblate/weblate/tags/> for full list of available ones.

Tag name	Descripción	Caso de uso
latest	Weblate stable release, matches latest tagged release	Rolling updates in a production environment
<VERSION>-<PLATFORM>	Weblate stable release	Well defined deploy in a production environment
edge	Weblate stable release with development changes in the Docker container (for example updated dependencies)	Rolling updates in a staging environment
edge-<DATE>-<PLATFORM>	Weblate stable release with development changes in the Docker container (for example updated dependencies)	Well defined deploy in a staging environment
bleeding	Development version Weblate from Git	Rolling updates to test upcoming Weblate features
bleeding-<DATE>-<PLATFORM>	Development version Weblate from Git	Well defined deploy to test upcoming Weblate features

Every image is tested by our CI before it gets published, so even the *bleeding* version should be quite safe to use.

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>;
```

(continué en la próxima página)

(proviene de la página anterior)

```

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

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

```

Replace <SITE_URL>, <SITE> and <EXPOSED_DOCKER_PORT> with actual values from your environment.

Certificados SSL automáticos con Let's Encrypt

In case you want to use [Let's Encrypt](#) automatically generated SSL certificates on public installation, you need to add a reverse HTTPS proxy an additional Docker container, [https-portal](#) will be used for that. This is made use of in the `docker-compose-https.yml` file. Then create a `docker-compose-https.override.yml` file with your settings:

```

version: '3'
services:
  weblate:
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for admin user
  https-portal:
    environment:
      DOMAINS: 'weblate.example.com -> http://weblate:8080'

```

Whenever invoking `docker-compose` you need to pass both files to it, and then do:

```

docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml
↩️ build
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml up

```

Actualizar el contenedor de Docker

Usually it is good idea to only update the Weblate container and keep the PostgreSQL container at the version you have, as upgrading PostgreSQL is quite painful and in most cases does not bring many benefits.

Distinto en la versión 4.10-1: Since Weblate 4.10-1, the Docker container uses Django 4.0 what requires PostgreSQL 10 or newer, please upgrade it prior to upgrading Weblate. See [Upgrade from 4.9 to 4.10](#) and [Upgrading PostgreSQL container](#).

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

```

# Fetch latest versions of the images
docker-compose pull
# Stop and destroy the containers
docker-compose down
# Spawn new containers in the background
docker-compose up -d

```

(continué en la próxima página)

(proviene de la página anterior)

```
# Follow the logs during upgrade
docker-compose logs -f
```

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

Nota: Upgrades across major versions are not supported by Weblate. For example, if you are on 3.x series and want to upgrade to 4.x, first upgrade to the latest 4.0.x-y image (at time of writing this it is the 4.0.4-5), which will do the migration and then continue upgrading to newer versions.

You might also want to update the `docker-compose` repository, though it's not needed in most case. See [Upgrading PostgreSQL container](#) for upgrading the PostgreSQL server.

Upgrading PostgreSQL container

PostgreSQL containers do not support automatic upgrading between version, you need to perform the upgrade manually. Following steps show one of the options of upgrading.

Ver también:

<https://github.com/docker-library/postgres/issues/37>

1. Stop Weblate container:

```
docker-compose stop weblate cache
```

2. Backup the database:

```
docker-compose exec database pg_dumpall --clean --username weblate > backup.sql
```

3. Stop the database container:

```
docker-compose stop database
```

4. Remove the PostgreSQL volume:

```
docker-compose rm -v database
docker volume remove weblate_postgres-data
```

5. Adjust `docker-compose.yml` to use new PostgreSQL version.

6. Start the database container:

```
docker-compose up -d database
```

7. Restore the database from the backup:

```
cat backup.sql | docker-compose exec -T database psql --username weblate --
↳ dbname postgres
```

8. Start all remaining containers:

```
docker-compose up -d
```

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:
  WEBLATE_WORKERS: 2
```

You can also fine-tune individual worker categories:

```
environment:
  WEB_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Ver también:

`WEBLATE_WORKERS`, `CELERY_MAIN_OPTIONS`, `CELERY_NOTIFY_OPTIONS`, `CELERY_MEMORY_OPTIONS`, `CELERY_TRANSLATE_OPTIONS`, `CELERY_BACKUP_OPTIONS`, `CELERY_BEAT_OPTIONS`, `WEB_WORKERS`

Scaling horizontally

Nuevo en la versión 4.6.

You can run multiple Weblate containers to scale the service horizontally. The `/app/data` volume has to be shared by all containers, it is recommended to use cluster filesystem such as GlusterFS for this. The `/app/cache` volume should be separate for each container.

Each Weblate container has defined role using `WEBLATE_SERVICE` environment variable. Please follow carefully the documentation as some of the services should be running just once in the cluster and the ordering of the services matters as well.

You can find example setup in the `docker-compose` repo as `docker-compose-split.yml`.

Docker environment variables

Many of Weblate's *Configuración* can be set in the Docker container using 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_LOGLEVEL_DATABASE

Configures the logging of the database queries verbosity.

WEBLATE_SITE_TITLE

Modifica el título del sitio que se muestra en la cabecera de todas las páginas.

WEBLATE_SITE_DOMAIN

Configura el dominio del sitio. Este parámetro es obligatorio.

Ver también:

Set correct site domain, SITE_DOMAIN

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Configures the site-admin's name and e-mail. It is used for both *ADMINS* setting and creating *admin* user (see *WEBLATE_ADMIN_PASSWORD* for more info on that).

Ejemplo:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

Ver también:

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_PASSWORD_FILE*, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

WEBLATE_ADMIN_PASSWORD_FILE

Sets the path to a file containing the password for the *admin* user.

Ver también:

WEBLATE_ADMIN_PASSWORD

WEBLATE_SERVER_EMAIL

The email address that error messages are sent from.

Ver también:

SERVER_EMAIL, *Configure e-mail sending*

WEBLATE_DEFAULT_FROM_EMAIL

Configures the address for outgoing e-mails.

Ver también:

DEFAULT_FROM_EMAIL, *Configure e-mail sending*

WEBLATE_CONTACT_FORM

Configures contact form behavior, see *CONTACT_FORM*.

WEBLATE_ALLOWED_HOSTS

Configures allowed HTTP hostnames using *ALLOWED_HOSTS*.

El valor predeterminado es *, que permite todos los nombres de anfitrión.

Ejemplo:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com,example.com
```

Ver también:

ALLOWED_HOSTS, *Allowed hosts setup*, *Set correct site domain*

WEBLATE_REGISTRATION_OPEN

Configures whether registrations are open by toggling *REGISTRATION_OPEN*.

Ejemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Configure which authentication methods can be used to create new account via *REGISTRATION_ALLOW_BACKENDS*.

Ejemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
  ↪oauth2
```

WEBLATE_TIME_ZONE

Configura el huso horario utilizado en Weblate; vea *TIME_ZONE*.

Nota: Para cambiar el huso horario del contenedor Docker, utilice la variable de entorno TZ.

Ejemplo:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Makes Weblate assume it is operated behind a reverse HTTPS proxy, it makes Weblate use HTTPS in e-mail and API links or set secure flags on cookies.

Consejo: Please see [ENABLE_HTTPS](#) documentation for possible caveats.

Nota: Esto no hace que el contenedor de Weblate acepte las conexiones HTTPS; debe configurarlas también. Vea *Contenedor Docker con compatibilidad con HTTPS* para obtener ejemplos.

Ejemplo:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Ver también:

[ENABLE_HTTPS](#) Set correct site domain, [WEBLATE_SECURE_PROXY_SSL_HEADER](#)

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

Nuevo en la versión 4.12.1.

Lets Weblate set the *meta[name=monetization]* field in the head of the document. If multiple are specified, chooses one randomly.

Ver también:

[INTERLEDGER_PAYMENT_POINTERS](#)

WEBLATE_IP_PROXY_HEADER

Permite que Weblate recupere la dirección IP de cualquier cabecera HTTP que se indique. Utilice esta variable si usa un «proxy» inverso ante el contenedor de Weblate.

Enables [IP_BEHIND_REVERSE_PROXY](#) and sets [IP_PROXY_HEADER](#).

Nota: The format must conform to Django's expectations. Django [transforms](#) raw HTTP header names as follows:

- convierte todas las letras en mayúsculas
- sustituye cualquier guion por guiones bajos
- antepone el prefijo HTTP_

So X-Forwarded-For would be mapped to HTTP_X_FORWARDED_FOR.

Ejemplo:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

A tuple representing a HTTP header/value combination that signifies a request is secure. This is needed when Weblate is running behind a reverse proxy doing SSL termination which does not pass standard HTTPS headers.

Ejemplo:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

Ver también:

[SECURE_PROXY_SSL_HEADER](#)

WEBLATE_REQUIRE_LOGIN

Enables [REQUIRE_LOGIN](#) to enforce authentication on whole Weblate.

Ejemplo:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS**WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS****WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS**

Adds URL exceptions for authentication required for the whole Weblate installation using [LOGIN_REQUIRED_URLS_EXCEPTIONS](#).

You can either replace whole settings, or modify default value using ADD and REMOVE variables.

WEBLATE_GOOGLE_ANALYTICS_ID

Configures ID for Google Analytics by changing [GOOGLE_ANALYTICS_ID](#).

WEBLATE_GITHUB_USERNAME

Configures GitHub username for GitHub pull-requests by changing [GITHUB_USERNAME](#).

Ver también:

[GitHub pull requests](#)

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 pull requests](#)

WEBLATE_GITLAB_USERNAME

Configures GitLab username for GitLab merge-requests by changing [GITLAB_USERNAME](#)

Ver también:

[GitLab merge requests](#)

WEBLATE_GITLAB_TOKEN

Configures GitLab personal access token for GitLab merge-requests via API by changing [GITLAB_TOKEN](#)

Ver también:

[GitLab merge requests](#)

WEBLATE_PAGURE_USERNAME

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

Ver también:

Pagure merge requests

WEBLATE_PAGURE_TOKEN

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

Ver también:

Pagure merge requests

WEBLATE_DEFAULT_PULL_MESSAGE

Configures the default title and message for pull requests via API by changing *DEFAULT_PULL_MESSAGE*

Ver también:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

Configures the language simplification policy, see *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configures the default *Control de acceso* for new projects, see *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configures the default value for *Acceso restringido* for new components, see *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configures the default value for *Permitir propagación de traducciones* for new components, see *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Configura *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Configura *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

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*

WEBLATE_API_RATELIMIT_ANON**WEBLATE_API_RATELIMIT_USER**

Nuevo en la versión 4.11.

Configures API rate limiting. Defaults to 100/day for anonymous and 5000/hour for authenticated users.

Ver también:

API rate limiting

WEBLATE_ENABLE_AVATARS

Nuevo en la versión 4.6.1.

Configures *ENABLE_AVATARS*.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Nuevo en la versión 4.9.

Configures *LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH*.

WEBLATE_SSH_EXTRA_ARGS

Nuevo en la versión 4.9.

Configures *SSH_EXTRA_ARGS*.

WEBLATE_BORG_EXTRA_ARGS

Nuevo en la versión 4.9.

Configures *BORG_EXTRA_ARGS*.

Configuración de traducción automática

Consejo: Configuring API key for a service automatically configures it in *MT_SERVICES*.

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_URL

Configures *DeepL* API version to use, see *MT_DEEPL_API_URL*.

WEBLATE_MT_LIBRETRANSLATE_KEY

Enables *LibreTranslate* machine translation and sets *MT_LIBRETRANSLATE_KEY*

WEBLATE_MT_LIBRETRANSLATE_API_URL

Configures *LibreTranslate* API instance to use, see *MT_LIBRETRANSLATE_API_URL*.

WEBLATE_MT_GOOGLE_KEY

Activa *Google Translate* y establece *MT_GOOGLE_KEY*

WEBLATE_MT_GOOGLE_CREDENTIALS

Enables *Google Translate API V3 (Advanced)* and sets *MT_GOOGLE_CREDENTIALS*

WEBLATE_MT_GOOGLE_PROJECT

Enables *Google Translate API V3 (Advanced)* and sets *MT_GOOGLE_PROJECT*

WEBLATE_MT_GOOGLE_LOCATION

Enables *Google Translate API V3 (Advanced)* and sets *MT_GOOGLE_LOCATION*

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
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH_UNION: ou=users,dc=example,
  ↪dc=com|ou=otherusers,dc=example,dc=com
```

Example with search and bind against Active Directory:

```
environment:
  WEBLATE_AUTH_LDAP_BIND_DN: CN=ldap,CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_BIND_PASSWORD: password
  WEBLATE_AUTH_LDAP_SERVER_URI: ldap://ldap.example.org
  WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS: 0
  WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
  WEBLATE_AUTH_LDAP_USER_SEARCH: CN=Users,DC=example,DC=com
  WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER: (sAMAccountName=%(user)s)
```

Ver también:

Autenticación LDAP

GitHub

WEBLATE_SOCIAL_AUTH_GITHUB_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID

Activa la *Autenticación por GitHub*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Activa la *Autenticación por Bitbucket*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Activa la *OAuth 2 de Facebook*.

Google

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS

Activa la *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

Disables e-mail authentication when set to any value. See [Desactivar la autenticación por contraseña](#).

Puesta en marcha de la base de datos PostgreSQL

The database is created by `docker-compose.yml`, so these settings affect both Weblate and PostgreSQL containers.

Ver también:

[Configuración de base de datos para Weblate](#)

POSTGRES_PASSWORD

Contraseña de PostgreSQL.

POSTGRES_PASSWORD_FILE

Path to the file containing the PostgreSQL password. Use as an alternative to `POSTGRES_PASSWORD`.

POSTGRES_USER

Nombre de usuario de PostgreSQL.

POSTGRES_DATABASE

Nombre de base de datos de PostgreSQL.

POSTGRES_HOST

PostgreSQL server hostname or IP address. Defaults to `database`.

POSTGRES_PORT

PostgreSQL server port. Defaults to none (uses the default value).

POSTGRES_SSL_MODE

Configure how PostgreSQL handles SSL in connection to the server, for possible choices see [SSL Mode Descriptions](#)

POSTGRES_ALTER_ROLE

Configures name of role to alter during migrations, see [Configurar Weblate para que utilice PostgreSQL](#).

POSTGRES_CONN_MAX_AGE

Nuevo en la versión 4.8.1.

The lifetime of a database connection, as an integer of seconds. Use 0 to close database connections at the end of each request (this is the default behavior).

Enabling connection persistence will typically, cause more open connection to the database. Please adjust your database configuration prior enabling.

Ejemplo de configuración:

```
environment:
  POSTGRES_CONN_MAX_AGE: 3600
```

Ver también:

[CONN_MAX_AGE](#), [Persistent connections](#)

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

Nuevo en la versión 4.9.1.

Disable server side cursors in the database. This is necessary in some **pgbouncer** setups.

Ejemplo de configuración:

```
environment:
  POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Ver también:

[DISABLE_SERVER_SIDE_CURSORS](#), [Transaction pooling and server-side cursors](#)

Configuración de copia de respaldo de la base de datos

Ver también:

Datos volcados para las copias de respaldo

WEBLATE_DATABASE_BACKUP

Configures the daily database dump using [DATABASE_BACKUP](#). Defaults to `plain`.

Caching server setup

Using Redis is strongly recommended by Weblate and you have to provide a Redis instance when running Weblate in Docker.

Ver también:

Enable caching

REDIS_HOST

The Redis server hostname or IP address. Defaults to `cache`.

REDIS_PORT

The Redis server port. Defaults to `6379`.

REDIS_DB

The Redis database number, defaults to `1`.

REDIS_PASSWORD

La contraseña del servidor Redis, no utilizada de manera predeterminada.

REDIS_TLS

Permite el uso de SSL para la conexión con Redis.

REDIS_VERIFY_SSL

Se puede utilizar para desactivar la verificación de certificados SSL para la conexión con Redis.

Puesta en funcionamiento del servidor de correo

Para que funcione el correo saliente, debe proporcionar un servidor de correo.

Ejemplo de configuración de TLS:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Ejemplo de configuración de SSL:

```
environment:
  WEBLATE_EMAIL_HOST: smtp.example.com
  WEBLATE_EMAIL_PORT: 465
  WEBLATE_EMAIL_HOST_USER: user
  WEBLATE_EMAIL_HOST_PASSWORD: pass
  WEBLATE_EMAIL_USE_TLS: 0
  WEBLATE_EMAIL_USE_SSL: 1
```

Ver también:

Configurar el correo electrónico saliente

WEBLATE_EMAIL_HOST

Nombre de anfitrión o dirección IP del servidor de correo.

Ver también:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_SSL`, `WEBLATE_EMAIL_USE_TLS`,
`EMAIL_HOST`

WEBLATE_EMAIL_PORT

Puerto del servidor de correo, cuyo valor predeterminado es 25.

Ver también:

`EMAIL_PORT`

WEBLATE_EMAIL_HOST_USER

Usuario de autenticación del correo electrónico.

Ver también:

`EMAIL_HOST_USER`

WEBLATE_EMAIL_HOST_PASSWORD

Contraseña de autenticación del correo electrónico.

Ver también:

`EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Path to the file containing the e-mail authentication password.

Ver también:

`WEBLATE_EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_USE_SSL

Whether to use an implicit TLS (secure) connection when talking to the SMTP server. In most e-mail documentation, this type of TLS connection is referred to as SSL. It is generally used on port 465. If you are experiencing problems, see the explicit TLS setting *WEBLATE_EMAIL_USE_TLS*.

Distinto en la versión 4.11: The SSL/TLS support is automatically enabled based on the *WEBLATE_EMAIL_PORT*.

Ver también:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_TLS*, *EMAIL_USE_SSL*

WEBLATE_EMAIL_USE_TLS

Whether to use a TLS (secure) connection when talking to the SMTP server. This is used for explicit TLS connections, generally on port 587 or 25. If you are experiencing connections that hang, see the implicit TLS setting *WEBLATE_EMAIL_USE_SSL*.

Distinto en la versión 4.11: The SSL/TLS support is automatically enabled based on the *WEBLATE_EMAIL_PORT*.

Ver también:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, *EMAIL_USE_TLS*

WEBLATE_EMAIL_BACKEND

Configura el dorsal Django para utilizarlo para enviar mensajes de correo electrónico.

Ver también:

Configure e-mail sending, *EMAIL_BACKEND*

WEBLATE_AUTO_UPDATE

Configures if and how Weblate should update repositories.

Ver también:

AUTO_UPDATE

Nota: This is a Boolean setting (use "true" or "false").

Integración del sitio

WEBLATE_GET_HELP_URL

Configura *GET_HELP_URL*.

WEBLATE_STATUS_URL

Configura *STATUS_URL*.

WEBLATE_LEGAL_URL

Configura *LEGAL_URL*.

WEBLATE_PRIVACY_URL

Configures *PRIVACY_URL*.

Informe de defectos

Se recomienda recopilar sistemáticamente los errores que se producen en la instalación; vea ref:*collecting-errors*.

Para activar la compatibilidad con Rollbar, defina lo siguiente:

ROLLBAR_KEY

Su ficha de acceso POST al servidor de Rollbar.

ROLLBAR_ENVIRONMENT

Su entorno de Rollbar, cuyo valor predeterminado es `production`.

Para activar la compatibilidad con Sentry, defina lo siguiente:

SENTRY_DSN

Su DSN de Sentry.

SENTRY_ENVIRONMENT

Su entorno de Sentry (opcional).

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`

Changing enabled apps, checks, add-ons or autofixes

Nuevo en la versión 3.8-5.

The built-in configuration of enabled checks, add-ons or autofixes can be adjusted by the following variables:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Ejemplo:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
  ↪ SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon, customize.addons.OtherAddon
```

Ver también:

CHECK_LIST, AUTOFIX_LIST, WEBLATE_ADDONS, INSTALLED_APPS

Configuración de contenedor

WEBLATE_WORKERS

Nuevo en la versión 4.6.1.

Base number of worker processes running in the container. When not set it is determined automatically on container startup based on number of CPU cores available.

It is used to determine *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS*, and *WEB_WORKERS*. You can use these settings to fine-tune.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

These variables allow you to adjust Celery worker options. It can be useful to adjust concurrency (`--concurrency 16`) or use different pool implementation (`--pool=gevent`).

By default, the number of concurrent workers is based on *WEBLATE_WORKERS*.

Ejemplo:

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

Ver también:

Celery worker options, Tareas en segundo plano con Celery

WEB_WORKERS

Configure how many uWSGI workers should be executed.

It defaults to *WEBLATE_WORKERS*.

Ejemplo:

```
environment:
  WEB_WORKERS: 32
```


WEBLATE_SERVICE

Defines which services should be executed inside the container. Use this for *Scaling horizontally*.

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 Servidor web.

Volúmenes de contenedores Docker

There are two volumes (data and cache) exported by the Weblate container. The other service containers (PostgreSQL or Redis) have their data volumes as well, but those are not covered by this document.

The data volume is used to store Weblate persistent data such as cloned repositories or to customize Weblate installation.

The placement of the Docker volume on host system depends on your Docker configuration, but usually it is stored in `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/` (the path consist of name of your docker-compose directory, container, and volume names). In the container it is mounted as `/app/data`.

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

When creating the volumes manually, the directories should be owned by UID 1000 as that is user used inside the container.

Ver también:

[Docker volumes documentation](#)

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.

This approach can be also used to override Weblate templates. For example *Información legal* documents can be placed into `/app/data/python/customize/templates/legal/documents`.

Alternatively you can also include own module (see *Personalizar Weblate*) and add it as separate volume to the Docker container, for example:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/weblate_
      ↪ customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Añadir módulos propios de Python

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

Configuring PostgreSQL server

The PostgreSQL container uses default PostgreSQL configuration and it won't effectively utilize your CPU cores or memory. It is recommended to customize the configuration to improve the performance.

The configuration can be adjusted as described in *Database Configuration* at https://hub.docker.com/_/postgres. The configuration matching your environment can be generated using <https://pgtune.leopard.in.ua/>.

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 \
  libffi-dev libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev \
  libacl1-dev libssl-dev libpq-dev libjpeg62-turbo-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 libleptonica-dev
apt install libldap2-dev libldap-common libsasl2-dev
apt install libxmlsec1-dev
```

Optionally install software for running production server, see *Running server*, *Configuración de base de datos para Weblate*, *Tareas en segundo plano con Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Las instrucciones de instalación local:

```
# Web server option 1: NGINX and uWSGI
apt install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
apt install apache2 libapache2-mod-wsgi-py3

# 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. Install Weblate including all optional dependencies:

```
# pkgconfig is needed to install borgbackup 1.2
pip install pkgconfig
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Please check [Dependencias opcionales](#) for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi_
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: 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 will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see [Adjusting configuration](#).
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check [Configuración de base de datos para Weblate](#) for 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/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **weblate check --deploy**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Instalar en SUSE y openSUSE

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 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-devel libffi-devel cairo-devel pango-devel \
  gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
  python3-devel git
```

Instale las dependencias opcionales que desee en función de las funcionalidades que vaya a utilizar (vea *Dependencias opcionales*):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libsasl2-devel
zypper install libxmlsec1-devel
```

Optionally install software for running production server, see *Running server*, *Configuración de base de datos para Weblate*, *Tareas en segundo plano con Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Las instrucciones de instalación local:

```
# Web server option 1: NGINX and uWSGI
zypper install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
zypper install apache2 apache2-mod_wsgi

# Caching backend: Redis
zypper install redis-server

# Database server: PostgreSQL
```

(continué en la próxima página)

(proviene de la página anterior)

```
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. Install Weblate including all optional dependencies:

```
# pkgconfig is needed to install borgbackup 1.2
pip install pkgconfig
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Please check [Dependencias opcionales](#) for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi_
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: 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 will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see [Adjusting configuration](#).

3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuración de base de datos para Weblate* for 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/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **weblate check --deploy**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Instalar en Red Hat, Fedora y CentOS

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 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-devel libffi-devel cairo-devel pango-devel \
    gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
    python3-devel git
```

Instale las dependencias opcionales que desee en función de las funcionalidades que vaya a utilizar (vea *Dependencias opcionales*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
dnf install libldap2-devel libsasl2-devel
dnf install libxmlsec1-devel
```

Optionally install software for running production server, see *Running server*, *Configuración de base de datos para Weblate*, *Tareas en segundo plano con Celery*. Depending on size of your installation you might want to run these components on dedicated servers.

Las instrucciones de instalación local:

```
# Web server option 1: NGINX and uWSGI
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
```

(continué en la próxima página)

(proviene de la página anterior)

```
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. Install Weblate including all optional dependencies:

```
# pkgconfig is needed to install borgbackup 1.2
pip install pkgconfig
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Please check [Dependencias opcionales](#) for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: 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 will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see [Adjusting configuration](#).

3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuración de base de datos para Weblate* for 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/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **weblate check --deploy**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Instalar en macOS

Requisitos de *hardware*

Weblate should run on any contemporary hardware without problems, the following is the minimal configuration required to run Weblate on a single host (Weblate, database and webserver):

- 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
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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. Install Weblate including all optional dependencies:

```
# pkgconfig is needed to install borgbackup 1.2
pip install pkgconfig
# Install Weblate with all optional dependencies
pip install "Weblate[all]"
```

Please check [Dependencias opcionales](#) for fine-tuning of optional dependencies.

Nota: On some Linux distributions running Weblate fails with libffi error:

```
ffi_prep_closure(): bad user_data (it seems that the version of the libffi
→library seen at runtime is different from the 'ffi.h' file seen at compile-
→time)
```

This is caused by incompatibility of binary packages distributed via PyPI with the distribution. To address this, you need to rebuild the package on your system:

```
pip install --force-reinstall --no-binary :all: cffi
```

Configurar Weblate

Nota: 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 will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see [Adjusting configuration](#).
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check [Configuración de base de datos para Weblate](#) for 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/`.
- Sign in with admin credentials obtained during installation or register with new users.
- You can now run Weblate commands using **weblate** command when Weblate virtualenv is active, see *Órdenes de gestión*.
- You can stop the test server with Ctrl+C.
- Review potential issues with your installation either on `/manage/performance/` URL (see *Interfaz de gestión*) or using **weblate check --deploy**, see *Puesta en marcha de entorno de producción*.

Adding translation

1. Open the admin interface (`http://localhost:8000/create/project/`) and create the project you want to translate. See *Configuración de proyectos* for more details.

All you need to specify here is the project name and its website.

2. Create a component which is the real object for translation - it points to the VCS repository, and selects which files to translate. See *Configuración de componentes* for more details.

The important fields here are: *Nombre de componente*, *Repositorio de código fuente*, and *Máscara de archivos* for finding translatable files. Weblate supports a wide range of formats including *gettext de GNU*, *Recursos de cadenas de Android*, *Cadenas de iOS de Apple*, *Propiedades de Java*, *Stringsdict format* or *Fluent format*, see *Formatos de archivo admitidos* for more details.

3. Once the above is completed (it can be lengthy process depending on the size of your VCS repository, and number of messages to translate), you can start translating.

Instalar desde el código fuente

1. 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 will need to provide at least the database credentials and Django secret key, but you will want more changes for production setup, see [Adjusting configuration](#).
6. Create the database used by Weblate, see [Configuración de base de datos para Weblate](#).
7. Build Django tables, static files and initial data (see [Filling up the database](#) and [Serving static files](#)):

```
weblate migrate
weblate collectstatic
weblate compress
```

Nota: Este paso debe repetirse siempre que actualice el repositorio.

Instalar en OpenShift

With the OpenShift Weblate template you can get your personal Weblate instance up and running in seconds. All of Weblate's dependencies are already included. PostgreSQL is set up as the default database and persistent volume claims are used.

You can find the template at <<https://github.com/WeblateOrg/openshift/>>.

Instalación

Los ejemplos siguientes presuponen que cuenta con un entorno OpenShift versión 3.x en funcionamiento y la herramienta de cliente `oc` instalada. Consulte la documentación de OpenShift para obtener instrucciones.

The `template.yml` is suited for running all components in OpenShift. There is also `template-external-postgresql.yml` which does not start a PostgreSQL server and allows you to configure external PostgreSQL server.

Consola web

Copy the raw content from [template.yml](#) and import them into your project, then use the **Create** button in the OpenShift web console to create your application. The web console will prompt you for the values for all of the parameters used by the template.

CLI

To upload the Weblate template to your current project's template library, pass the `template.yml` file with the following command:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
  -n <PROJECT>
```

The template is now available for selection using the web console or the CLI.

Parámetros

The parameters that you can override are listed in the parameters section of the template. You can list them with the CLI by using the following command and specifying the file to be used:

```
$ oc process --parameters -f https://raw.githubusercontent.com/WeblateOrg/
↪openshift/main/template.yml

# If the template is already uploaded
$ oc process --parameters -n <PROJECT> weblate
```

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
```

Configuración

For further configuration options, see *Docker environment variables* for full list of environment variables.

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

- *Instalar con Docker*, recomendable para montajes en entornos de producción.
- Instalación en entorno virtual, recomendable para montajes en entornos de producción:
 - *Instalar en Debian y Ubuntu*
 - *Instalar en SUSE y openSUSE*
 - *Instalar en Red Hat, Fedora y CentOS*
 - *Instalar en macOS*
- *Instalar desde el código fuente*, recomendable para el desarrollo.
- *Instalar en OpenShift*

- *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.celeryq.dev/>

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

translation-finder <https://github.com/WeblateOrg/translation-finder>

Python Social Auth <https://python-social-auth.readthedocs.io/>

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 (optional for *Mercurial repositories support*) <https://www.mercurial-scm.org/>

phply (optional for *Cadenas de PHP*) <https://github.com/viraptor/phply>

tesseract (optional for OCR in *Contexto visual para cadenas*) <https://github.com/sirfz/tesseract>

python-akismet (optional for *Spam protection*) <https://github.com/Nekmo/python-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/>

fluent.syntax (optional for *Fluent format*) <https://projectfluent.org/>

Consejo: When installing using pip, you can directly specify desired features when installing:

```
pip install "Weblate[PHP,Fluent]"
```

Or you can install Weblate with all optional features:

```
pip install "Weblate[all]"
```

Or you can install Weblate without any optional features:

```
pip install Weblate
```

Dependencias del motor de la base de datos

Weblate supports PostgreSQL, MySQL and MariaDB, see *Configuración de base de datos para Weblate* and backends documentation for more details.

Otros requisitos de sistema

Deben instalarse las dependencias siguientes en el sistema:

Git <https://git-scm.com/>

Pango, Cairo and related header files and GObject introspection data <https://cairographics.org/>, <https://pango.gnome.org/>, see *Pango y Cairo*

git-review (opcional para admitir Gerrit) <https://pypi.org/project/git-review/>

git-svn (opcional para admitir Subversion) <https://git-scm.com/docs/git-svn>

tesseract y sus datos (opcional para el reconocimiento óptico de caracteres en capturas de pantalla) <https://github.com/tesseract-ocr/tesseract>

licensee (optional for detecting license when creating component) <https://github.com/licensee/licensee>

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@braiins.cz>
sub  rsa4096 2009-06-17 [E]
sub  rsa4096 2015-09-09 [S]
```

- Descargue el llavero del [servidor de Michal](#) y, a continuación, impórtelo con:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Descargue e importe la clave de uno de los servidores de claves:

```
$ gpg --keyserver hkp://pgp.mit.edu --recv-keys 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: key 9C27B31342B7511D: "Michal Čihař <michal@cihar.com>" imported
gpg: Total number processed: 1
gpg:          unchanged: 1
```

This will improve the situation a bit - at this point you can verify that the signature from the given key is correct but you still can not trust the name used in the key:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:          using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:          aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:          aka "[jpeg image of size 8848]" [ultimate]
gpg:          aka "Michal Čihař (Braiiins) <michal.cihar@braiins.cz>" [ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
```

(continué en la próxima página)

(proviene de la página anterior)

```
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 (Brains) <michal.cihar@brains.cz>" [ultimate]
↪ [ultimate]
```

Should the signature be invalid (the archive has been changed), you would get a clear error regardless of the fact that the key is trusted or not:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:          using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: BAD signature from "Michal Čihař <michal@cihar.com>" [ultimate]
```

2.1.4 Permisos del sistema de archivos

The Weblate process needs to be able to read and write to the directory where it keeps data - *DATA_DIR*. All files within this directory should be owned and writable by the user running all Weblate processes (typically WSGI and Celery, see [Running server](#) and [Tareas en segundo plano con Celery](#)).

The default configuration places them in the same tree as the Weblate sources, however you might prefer to move these to a better location such as: `/var/lib/weblate`.

Weblate tries to create these directories automatically, but it will fail when it does not have permissions to do so.

You should also take care when running *Órdenes de gestión*, as they should be ran under the same user as Weblate itself is running, otherwise permissions on some files might be wrong.

In the Docker container, all files in the `/app/data` volume have to be owned by the `weblate` user inside the container (UID 1000).

Ver también:

Serving static files

2.1.5 Configuración de base de datos para Weblate

Es recomendable ejecutar Weblate con un servidor de bases de datos PostgreSQL.

Ver también:

Use a powerful database engine, Databases, Migrating from other databases to PostgreSQL

PostgreSQL

PostgreSQL is usually the best choice for Django-based sites. It's the reference database used for implementing Django database layer.

Nota: Weblate uses trigram extension which has to be installed separately in some cases. Look for `postgresql-contrib` or a similarly named package.

Ver también:

[PostgreSQL notes](#)

Crear una base de datos en PostgreSQL

Suele ser una buena idea ejecutar Weblate en su propia base de datos, en una cuenta de usuario separada:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Consejo: If you don't want to make the Weblate user a superuser in PostgreSQL, you can omit that. In that case you will have to perform some of the migration steps manually as a PostgreSQL superuser in schema Weblate will use:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm 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",
```

(continué en la próxima página)

(proviene de la página anterior)

```

# 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 emoyis).
- Configure the server with `innodb_large_prefix` to allow longer indices on text fields.
- Set the isolation level to `READ COMMITTED`.
- The SQL mode should be set to `STRICT_TRANS_TABLES`.

MySQL 8.x, MariaDB 10.5.x or newer have reasonable default configuration so that no server tweaking should be necessary and all what is needed can be configured on the client side.

El siguiente es un **`:archivo:/etc/my.cnf.d/server.cnf`** de ejemplo para un servidor con 8 GB de RAM. Esta configuración debería bastar en la mayoría de las instalaciones. MySQL y MariaDB poseen configuraciones que optimizarán el rendimiento de su servidor, las cuales no es necesario ajustar a menos que espere tener cantidades elevadas de usuarios accediendo al sistema en simultáneo. Consulte la documentación del proveedor de su base de datos para obtener más información al respecto.

It is absolutely critical to reduce issues when installing that the setting `innodb_file_per_table` is set properly and MySQL/MariaDB restarted before you start your Weblate install.

```

[mysqld]
character-set-server = utf8mb4
character-set-client = utf8mb4
collation-server = utf8mb4_unicode_ci

datadir=/var/lib/mysql

log-error=/var/log/mariadb/mariadb.log

innodb_large_prefix=1
innodb_file_format=Barracuda
innodb_file_per_table=1

```

(continué en la próxima página)

(proviene de la página anterior)

```
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.

Another thing to take care of is the `Host` header. It should match to whatever is configured as `SITE_DOMAIN`. Additional configuration might be needed in your reverse proxy (for example use `ProxyPreserveHost On` for Apache or `proxy_set_header Host $host;` with nginx).

Ver también:

Spam protection, Rate limiting, Registro de auditoría, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER, IP_PROXY_OFFSET, SECURE_PROXY_SSL_HEADER

HTTP proxy

Weblate does execute VCS commands and those accept proxy configuration from environment. The recommended approach is to define proxy settings in `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Ver también:

Proxy Environment Variables

2.1.7 Adjusting configuration

Ver también:

Configuración de muestra

Copy `weblate/settings_example.py` to `weblate/settings.py` and adjust it to match your setup. You will probably want to adjust the following options: `ADMINS`

List of site administrators to receive notifications when something goes wrong, for example notifications on failed merges, or Django errors.

Ver también:

`ADMINS`, *Properly configure admins*

`ALLOWED_HOSTS`

You need to set this to list the hosts your site is supposed to serve. For example:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternatively you can include wildcard:

```
ALLOWED_HOSTS = ["*"]
```

Ver también:

[ALLOWED_HOSTS](#), [WEBLATE_ALLOWED_HOSTS](#), [Allowed hosts setup](#)

SESSION_ENGINE

Configure how your sessions will be stored. In case you keep the default database backend engine, you should schedule: **weblate clearsessions** to remove stale session data from the database.

If you are using Redis as cache (see [Enable caching](#)) it is recommended to use it for sessions as well:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Ver también:

[Configuring the session engine](#), [SESSION_ENGINE](#)

DATABASES

Connectivity to database server, please check Django's documentation for more details.

Ver también:

[Configuración de base de datos para Weblate](#), [DATABASES](#), [Databases](#)

DEBUG

Disable this for any production server. With debug mode enabled, Django will show backtraces in case of error to users, when you disable it, errors will be sent per e-mail to ADMINS (see above).

Debug mode also slows down Weblate, as Django stores much more info internally in this case.

Ver también:

[DEBUG](#), [Desactivar el modo de depuración](#)

DEFAULT_FROM_EMAIL

E-mail sender address for outgoing e-mail, for example registration e-mails.

Ver también:

[DEFAULT_FROM_EMAIL](#)

SECRET_KEY

Key used by Django to sign some info in cookies, see [Clave secreta de Django](#) for more info.

Ver también:

[SECRET_KEY](#)

SERVER_EMAIL

E-mail used as sender address for sending e-mails to the administrator, for example notifications on failed merges.

Ver también:

[SERVER_EMAIL](#)

2.1.8 Filling up the database

After your configuration is ready, you can run `weblate 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.

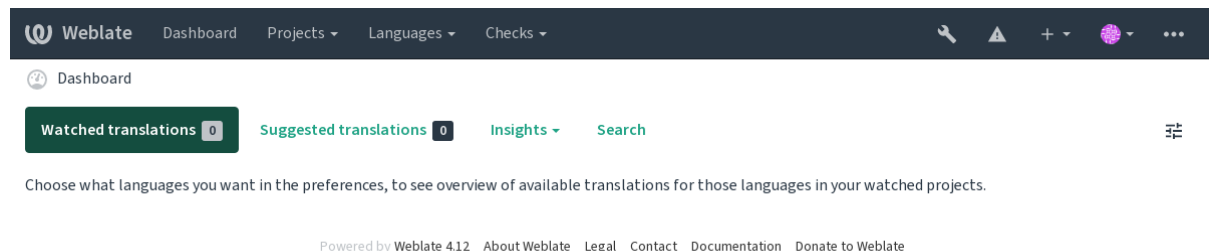
Once you are done, you should also check the *Performance report* in the admin interface, which will give you hints of potential non optimal configuration on your site.

Ver también:

Configuración, Lista de privilegios y roles incorporados

2.1.9 Puesta en marcha de entorno de producción

For a production setup you should carry out adjustments described in the following sections. The most critical settings will trigger a warning, which is indicated by an exclamation mark in the top bar if signed in as a superuser:



It is also recommended to inspect checks triggered by Django (though you might not need to fix all of them):

```
weblate check --deploy
```

You can also review the very same checklist from the *Interfaz de gestión*.

Ver también:

Deployment checklist

Desactivar el modo de depuración

Ejecute esto para desactivar el modo de depuración (*DEBUG*) de Django:

```
DEBUG = False
```

Con el modo de depuración activado, Django almacena todas las consultas ejecutadas y muestra a los usuarios el seguimiento regresivo de los errores, lo cual no es deseable en un entorno de producción.

Ver también:

Adjusting configuration

Properly configure admins

Set the correct admin addresses to the `ADMINS` setting to defining who will receive e-mails in case something goes wrong on the server, for example:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Ver también:

Adjusting configuration

Set correct site domain

Adjust site name and domain in the admin interface, otherwise links in RSS or registration e-mails will not work. This is configured using `SITE_DOMAIN` which should contain site domain name.

Distinto en la versión 4.2: Prior to the 4.2 release the Django sites framework was used instead, please see [The «sites» framework](#).

Ver también:

Allowed hosts setup, Correctly configure HTTPS `SITE_DOMAIN`, `WEBLATE_SITE_DOMAIN`, `ENABLE_HTTPS`

Correctly configure HTTPS

It is strongly recommended to run Weblate using the encrypted HTTPS protocol. After enabling it, you should set `ENABLE_HTTPS` in the settings:

```
ENABLE_HTTPS = True
```

Consejo: You might want to set up HSTS as well, see [SSL/HTTPS](#) for more details.

Ver también:

`ENABLE_HTTPS`, Allowed hosts setup, Set correct site domain

Set properly `SECURE_HSTS_SECONDS`

If your site is served over SSL, you have to consider setting a value for `SECURE_HSTS_SECONDS` in the `settings.py` to enable HTTP Strict Transport Security. By default it's set to 0 as shown below.

```
SECURE_HSTS_SECONDS = 0
```

If set to a non-zero integer value, the `django.middleware.security.SecurityMiddleware` sets the HTTP Strict Transport Security header on all responses that do not already have it.

Advertencia: Setting this incorrectly can irreversibly (for some time) break your site. Read the [HTTP Strict Transport Security](#) documentation first.

Use a powerful database engine

- Please use PostgreSQL for a production environment, see *Configuración de base de datos para Weblate* for more info.
- Use adjacent location for running the database server, otherwise the networking performance or reliability might ruin your Weblate experience.
- Check the database server performance or tweak its configuration, for example using PG Tune.

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,
    },
}
```

(continué en la próxima página)

(proviene de la página anterior)

```
"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": [
                (
                    "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. This varies between distribution and web servers, so check documentation of your web server packages for that.

Apache on Ubuntu uses `/etc/apache2/envvars`:

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

Apache on CentOS uses `/etc/sysconfig/httpd` (or `/opt/rh/httpd24/root/etc/sysconfig/httpd`):

```
LANG='en_US.UTF-8'
```

Using custom certificate authority

Weblate does verify SSL certificates during HTTP requests. In case you are using custom certificate authority which is not trusted in default bundles, you will have to add its certificate as trusted.

The preferred approach is to do this at system level, please check your distro documentation for more details (for example on debian this can be done by placing the CA certificate into `/usr/local/share/ca-certificates/` and running `update-ca-certificates`).

Once this is done, system tools will trust the certificate and this includes Git.

For Python code, you will need to configure requests to use system CA bundle instead of the one shipped with it. This can be achieved by placing following snippet to `settings.py` (the path is Debian specific):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```


Compressing client assets

Weblate comes with a bunch of JavaScript and CSS files. For performance reasons it is good to compress them before sending to a client. In default configuration this is done on the fly at cost of little overhead. On big installations, it is recommended to enable offline compression mode. This needs to be done in the configuration and the compression has to be triggered on every Weblate upgrade.

The configuration switch is simple by enabling `django.conf.settings.COMPRESS_OFFLINE` and configuring `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (the latter is already included in the example configuration):

```
COMPRESS_OFFLINE = True
```

On each deploy you need to compress the files to match current version:

```
weblate compress
```

Consejo: La imagen oficial para Docker ya tiene activada esta funcionalidad.

Ver también:

[Common Deployment Scenarios](#), [Serving static files](#)

2.1.10 Running server

Consejo: In case you are not experienced with services described below, you might want to try [Instalar con Docker](#).

Es necesario contar con varios servicios para ejecutar Weblate. El montaje recomendado consiste de:

- Servidor de base de datos (vea [Configuración de base de datos para Weblate](#))
- Servidor de antememoria (vea [Enable caching](#))
- Frontend web server for static files and SSL termination (see [Serving static files](#))
- Servidor WSGI para el contenido dinámico (vea [Sample configuration for NGINX and uWSGI](#))
- Celery para ejecutar las tareas en segundo plano (vea [Tareas en segundo plano con Celery](#))

Nota: There are some dependencies between the services, for example cache and database should be running when starting up Celery or uwsgi processes.

In most cases, you will run all services on single (virtual) server, but in case your installation is heavy loaded, you can split up the services. The only limitation on this is that Celery and Wsgi servers need access to `DATA_DIR`.

Nota: The WSGI process has to be executed under the same user the Celery process, otherwise files in the `DATA_DIR` will be stored with mixed ownership, leading to runtime issues.

See also [Permisos del sistema de archivos](#) and [Tareas en segundo plano con Celery](#).

Running web server

Running Weblate is not different from running any other Django based program. Django is usually executed as uWSGI or fcgi (see examples for different web servers 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*, *How to deploy Django*, *How to deploy static files*

Content security policy

The default Weblate configuration enables `weblate.middleware.SecurityMiddleware` middleware which sets security related HTTP headers like `Content-Security-Policy` or `X-XSS-Protection`. These are by default set up to work with Weblate and its configuration, but this might need customization for your environment.

Ver también:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

Sample configuration for NGINX and uWSGI

To run production webserver, use the wsgi wrapper installed with Weblate (in virtual env case it is installed as `~/weblate-env/lib/python3.7/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/python3.9/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_
↪request-timeout=600
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.7/site-packages/
↪weblate/wsgi.py process-group=weblate
    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>

```

(continué en la próxima página)

(proviene de la página anterior)

```

ServerAdmin admin@weblate.example.org
ServerName weblate.example.org

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

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

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

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

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

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

```

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/

```

(continué en la próxima página)

(proviene de la página anterior)

```

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

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

# Path to your Weblate virtualenv
WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate_
↪request-timeout=600
WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.7/site-packages/
↪weblate/wsgi.py process-group=weblate
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 execute regular and background tasks. You are supposed to run a Celery service that will execute these. For example, it is responsible for handling following operations (this list is not complete):

- Receiving webhooks from external services (see *Actuadores de notificación*).
- Running regular maintenance tasks such as backups, cleanups, daily add-ons, or updates (see *Respaldo y trasladar Weblate*, *BACKGROUND_TASKS*, *Complementos*).
- Running *Traducción automática*.
- Sending digest notifications.
- Offloading expensive operations from the wsgi process.
- Committing pending changes (see *Consignas diferidas*).

A typical setup using Redis as a backend looks like this:

```

CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

```

Ver también:

[Redis broker configuration in Celery](#)

You should also start the Celery worker to process the tasks and start scheduled tasks, this can be done directly on the command-line (which is mostly useful when debugging or developing):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Nota: The Celery process has to be executed under the same user as the WSGI process, otherwise files in the `DATA_DIR` will be stored with mixed ownership, leading to runtime issues.

See also *Permisos del sistema de archivos* and *Running server*.

Executing Celery tasks in the wsgi using eager mode

Nota: This will have severe performance impact on the web interface, and will break features depending on regular trigger (for example committing pending changes, digest notifications, or backups).

For development, you might want to use eager configuration, which does process all tasks in place:

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

Running Celery as system service

Most likely you will want to run Celery as a daemon and that is covered by [Daemonization](#). For the most common Linux setup using systemd, you can use the example files shipped in the `examples` folder listed below.

Systemd unit to be placed as `/etc/systemd/system/celery-weblate.service`:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target

[Service]
Type=forking
User=weblate
Group=weblate
EnvironmentFile=/etc/default/celery-weblate
WorkingDirectory=/home/weblate
RuntimeDirectory=celery
RuntimeDirectoryPreserve=restart
LogsDirectory=celery
ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES} \
  --pidfile=${CELERYD_PID_FILE}'
ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'

[Install]
WantedBy=multi-user.target
```

Environment configuration to be placed as `/etc/default/celery-weblate`:


```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-multiplier:celery=4 \
--queues:notify=notify --prefetch-multiplier:notify=10 \
--queues:memory=memory --prefetch-multiplier:memory=10 \
--queues:translate=translate --prefetch-multiplier:translate=4 \
--concurrency:backup=1 --queues:backup=backup --prefetch-multiplier:backup=2"

# Logging configuration
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
# and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"
```

Additional configuration to rotate Celery logs using **logrotate** to be placed as `/etc/logrotate.d/celery`:

```
/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}
```

Periodic tasks using Celery beat

Weblate comes with built-in setup for scheduled tasks. You can however define additional tasks in `settings.py`, for example see *Consignas diferidas*.

The tasks are supposed to be executed by Celery beats daemon. In case it is not working properly, it might not be running or its database was corrupted. Check the Celery startup logs in such case to figure out root cause.

Monitoring Celery status

You can find current length of the Celery task queues in the *Interfaz de gestión* or you can use `celery_queues` on the command-line. In case the queue will get too long, you will also get configuration error in the admin interface.

Advertencia: The Celery errors are by default only logged into Celery log and are not visible to user. In case you want to have overview on such failures, it is recommended to configure *Collecting error reports*.

Ver también:

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. The Docker container has built-in health check using this URL.

For monitoring metrics of Weblate you can use `GET /api/metrics/` API endpoint.

Ver también:

[Weblate plugin for Munin](#)

2.1.13 Collecting error reports

Weblate, as any other software, can fail. In order to collect useful failure states we recommend to use third party services to collect such information. This is especially useful in case of failing Celery tasks, which would otherwise only report error to the logs and you won't get notified on them. Weblate has support for the following services:

Sentry

Weblate has built-in support for [Sentry](#). To use it, it's enough to set `SENTRY_DSN` in the `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

Weblate has built-in support for [Rollbar](#). To use it, it's enough to follow instructions for [Rollbar notifier for Python](#).

In short, you need to adjust `settings.py`:

```
# Add rollbar as last middleware:
MIDDLEWARE = [
    # ... other middleware classes ...
    "rollbar.contrib.django.middleware.RollbarNotifierMiddleware",
]

# Configure client access
ROLLBAR = {
    "access_token": "POST_SERVER_ITEM_ACCESS_TOKEN",
    "client_token": "POST_CLIENT_ITEM_ACCESS_TOKEN",
    "environment": "development" if DEBUG else "production",
    "branch": "main",
    "root": "/absolute/path/to/code/root",
}
```

Everything else is integrated automatically, you will now collect both server and client side errors.

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.

Utilice su interfaz administrativa o pulse en este botón (que le llevará a su servidor):



También se puede utilizar la interfaz de línea de órdenes:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Actualizar Weblate

2.3.1 Docker image upgrades

The official Docker image (see [Instalar con Docker](#)) has all Weblate upgrade steps integrated. There are typically no manual steps needed besides pulling latest version.

Ver también:

[Actualizar el contenedor de Docker](#)

2.3.2 Instrucciones de actualización genéricas

Before upgrading, please check the current [Requisitos de software](#) as they might have changed. Once all requirements are installed or updated, please adjust your `settings.py` to match changes in the configuration (consult `settings_example.py` for correct values).

Always check [Instrucciones específicas de las versiones](#) before upgrade. In case you are skipping some versions, please follow instructions for all versions you are skipping in the upgrade. Sometimes it's better to upgrade to some intermediate version to ensure a smooth migration. Upgrading across multiple releases should work, but is not as well tested as single version upgrades.

Nota: It is recommended to perform a full database backup prior to upgrade so that you can roll back the database in case upgrade fails, see [Respaldar y trasladar Weblate](#).

1. Stop wsgi and Celery processes. The upgrade can perform incompatible changes in the database, so it is always safer to avoid old processes running while upgrading.
2. Upgrade Weblate code.

For pip installs it can be achieved by:

```
pip install -U "Weblate[all]"
```

If you don't want to install all of the optional dependencies do:

```
pip install -U Weblate
```

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

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

3. New Weblate release might have new *Dependencias opcionales*, please check if they cover features you want.
4. Upgrade configuration file, refer to `settings_example.py` or *Instrucciones específicas de las versiones* for needed steps.
5. Upgrade database structure:

```
weblate migrate --noinput
```

6. Collect updated static files (see *Running server* and *Serving static files*):

```
weblate collectstatic --noinput --clear
```

7. Compress JavaScript and CSS files (optional, see *Compressing client assets*):

```
weblate compress
```

8. If you are running version from Git, you should also regenerate locale files every time you are upgrading. You can do this by invoking:

```
weblate compilemessages
```

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

```
weblate check --deploy
```

10. 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 `MT_DEEPL_API_VERSION` setting has been removed in Version 4.7. The *DeepL* machine translation now uses the new `MT_DEEPL_API_URL` instead. You might need to adjust `MT_DEEPL_API_URL` to match your subscription.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.1 to 4.2

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- Upgrade from 3.x releases is not longer supported, please upgrade to 4.0 or 4.1 first.
- There are some new and updated requirements.
- There are several changes in `settings_example.py`, most notable new middleware and changed application ordering.
- The keys for JSON based formats no longer include leading dot. The strings are adjusted during the database migration, but external components might need adjustment in case you rely on keys in exports or API.
- The Celery configuration was changed to no longer use `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.
- The Weblate domain is now configured in the settings, see `SITE_DOMAIN` (or `WEBLATE_SITE_DOMAIN`). You will have to configure it before running Weblate.
- The username and email fields on user database now should be case insensitive unique. It was mistakenly not enforced with PostgreSQL.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.2 to 4.3

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- There are some changes in quality checks, you might want to include them in case you modified the `CHECK_LIST`.
- The source language attribute was moved from project to a component what is exposed in the API. You will need to update *Cliente de Weblate* in case you are using it.
- The database migration to 4.3 might take long depending on number of strings you are translating (expect around one hour of migration time per 100,000 source strings).
- There is a change in `INSTALLED_APPS`.
- There is a new setting `SESSION_COOKIE_AGE_AUTHENTICATED` which complements `SESSION_COOKIE_AGE`.
- In case you were using **hub** or **lab** to integrate with GitHub or GitLab, you will need to reconfigure this, see `GITHUB_CREDENTIALS` and `GITLAB_CREDENTIALS`.

Distinto en la versión 4.3.1:

- The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

Distinto en la versión 4.3.2:

- The `post_update` method of add-ons now takes extra `skip_push` parameter.

Ver también:

Instrucciones de actualización genéricas

Actualizar de la 4.3 a la 4.4

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- There is a change in `INSTALLED_APPS`, `weblate.configuration` has to be added there.
- Ahora es obligatorio el uso de Django 3.1.
- In case you are using MySQL or MariaDB, the minimal required versions have increased, see *MySQL y MariaDB*.

Distinto en la versión 4.4.1:

- *gettext monolingüe* now uses both `msgid` and `msgctxt` when present. This will change identification of translation strings in such files breaking links to Weblate extended data such as screenshots or review states. Please make sure you commit pending changes in such files prior upgrading and it is recommended to force loading of affected component using `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

Upgrade from 4.6 to 4.7

Please follow *Instrucciones de actualización genéricas* in order to perform update.

Notable configuration or dependencies changes:

- There are several changes in `settings_example.py`, most notable middleware changes (`MIDDLEWARE`), please adjust your settings accordingly.
- The *DeepL* machine translation now has a generic `MT_DEEPL_API_URL` setting to adapt to different subscription models more flexibly. The `MT_DEEPL_API_VERSION` setting is no longer used.
- Django 3.2 is now required.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.7 to 4.8

Please follow *Instrucciones de actualización genéricas* in order to perform update.

There are no additional upgrade steps needed in this release.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.8 to 4.9

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- There is a change in storing metrics, the upgrade can take long time on larger sites.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.9 to 4.10

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- There is a change in per-project groups, the upgrade can take long time on sites with thousands of projects.
- Django 4.0 has made some incompatible changes, see [Backwards incompatible changes in 4.0](#). Weblate still supports Django 3.2 for now, in case any of these are problematic. Most notable changes which might affect Weblate:
 - Dropped support for PostgreSQL 9.6, Django 4.0 supports PostgreSQL 10 and higher.
 - Format of `CSRF_TRUSTED_ORIGINS` was changed.
- The Docker container now uses Django 4.0, see above for changes.

Ver también:

Instrucciones de actualización genéricas

Upgrade from 4.10 to 4.11

Please follow *Instrucciones de actualización genéricas* in order to perform update.

- Weblate now requires Python 3.7 or newer.
- The implementation of *Gestionar el control de acceso por proyecto* has changed, removing the project prefix from the group names. This affects API users.
- Weblate now uses `charset-normalizer` instead of `chardet` module for character set detection.
- **Changed in 4.11.1:** There is a change in `REST_FRAMEWORK` setting (removal of one of the backends in `DEFAULT_AUTHENTICATION_CLASSES`).

Ver también:

Instrucciones de actualización genéricas

2.3.4 Upgrading from Python 2 to Python 3

Weblate no longer supports Python older than 3.6. In case you are still running on older version, please perform migration to Python 3 first on existing version and upgrade later. See [Upgrading from Python 2 to Python 3](#) in the Weblate 3.11.1 documentation.

2.3.5 Migrating from other databases to PostgreSQL

If you are running Weblate on other database than PostgreSQL, you should consider migrating to PostgreSQL as Weblate performs best with it. The following steps will guide you in migrating your data between the databases. Please remember to stop both web and Celery servers prior to the migration, otherwise you might end up with inconsistent data.

Crear una base de datos en PostgreSQL

Suele ser una buena idea ejecutar Weblate en su propia base de datos, en una cuenta de usuario separada:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Migrating using Django JSON dumps

The simplest approach for migration is to utilize Django JSON dumps. This works well for smaller installations. On bigger sites you might want to use pgloader instead, see [Migrar a PostgreSQL con pgloader](#).

1. Add PostgreSQL as additional database connection to the `settings.py`:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # 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


[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/tmpsyn1mjxweblate

Backup service credentials

Backup repository

/tmp/tmpsyn1mjxweblate

Passphrase

s^LHN^Y!S^OpF0tA&pjSM\$*P%4LZj^N(feSUMB4pLFfyImoz

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

April 20, 2022

Backup performed

April 20, 2022

Repository initialization

April 20, 2022

Turn off

Perform backup

Delete

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Add backup service

Backup repository URL

Use /path/to/repo for local backups or user@host:/path/to/repo or ssh://user@host:port/path/to/backups for remote SSH backups.

Add

Powered by Weblate 4.12
[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`

Customizing backup

- The database backup can be configured via `DATABASE_BACKUP`.
- The backup creation can be customized using `BORG_EXTRA_ARGS`.

2.4.2 Weblate provisioned backup storage

The easiest way of backing up your Weblate instance is purchasing the backup service at weblate.org. 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

For creating remote backups, you will have to install [BorgBackup](#) onto another server that's accessible for your Weblate deployment via SSH using the Weblate SSH key:

1. Prepare a server where your backups will be stored.
2. Install the SSH server on it (you will get it by default with most Linux distributions).
3. Install [BorgBackup](#) on that server; most Linux distributions have packages available (see [Installation](#)).
4. Choose an existing user or create a new user that will be used for backing up.
5. Add Weblate SSH key to the user so that Weblate can SSH to the server without a password (see [Clave SSH de Weblate](#)).
6. Configure the backup location in Weblate as `user@host:/path/to/backups` or `ssh://user@host:port/path/to/backups`.

Consejo: [Weblate provisioned backup storage](#) provides you automated remote backups without any effort.

Ver también:

[Clave SSH de Weblate, General](#)

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](#).

When using Docker container, the settings file is already included in the container and you should restore the original environment variables. The `environment.yml` file might help you with this (see [Datos volcados para las copias de respaldo](#)).

6. Copy the whole restored data dir to the location configured by `DATA_DIR`.

When using Docker container place the data into the data volume, see [Volúmenes de contenedores Docker](#).

Please make sure the files have correct ownership and permissions, see [Permisos del sistema de archivos](#).

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

Distinto en la versión 4.7: The environment dump was added as `environment.yml` to help in restoring in the Docker environments.

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`.
- Environment dump as `environment.yml`.

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

To restore the database backup load it using database tools, for example:

```
psql --file=database.sql weblate
```

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 *Enviar al consignar* 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

La configuración por defecto de Weblate es utilizar python-social-auth, un formulario en el sitio web para gestionar el registro de nuevos usuarios. Después de confirmar su correo electrónico, un nuevo usuario puede contribuir o autenticarse utilizando uno de los servicios de terceros.

También puedes desactivar el registro de nuevos usuarios mediante `REGISTRATION_OPEN`.

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

2.5.2 Dorsales de autenticación

La solución incorporada de Django se utiliza para la autenticación, incluyendo varias opciones sociales para hacerlo. Su uso permite importar la base de datos de usuarios de otros proyectos basados en Django (ver *Migrar de Pootle*).

Django también se puede configurar para configurar otros medios.

Ver también:

Configuración de autenticación describe cómo configurar la autenticación en la imagen oficial para Docker.

2.5.3 Autenticación social

Gracias a [Welcome to Python Social Auth's documentation!](#), Weblate admite la autenticación a través de numerosos servicios de terceros, tales como GitLab, Ubuntu y Fedora, entre otros.

Consulta su documentación para obtener instrucciones de configuración genéricas en [Django Framework](#).

Nota: Por defecto, Weblate confía en los servicios de autenticación de terceros para proporcionar una dirección de correo electrónico validada. Si algunos de los servicios que deseas utilizar no lo soportan, por favor, ejecuta la validación del correo electrónico en el lado de Weblate configurando `FORCE_EMAIL_VALIDATION` para ellos. Por ejemplo:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Ver también:

[Pipeline](#)

Habilitar backends individuales es bastante fácil, sólo es cuestión de añadir una entrada al ajuste `AUTHENTICATION_BACKENDS` y posiblemente añadir las claves necesarias para un método de autenticación determinado. Ten en cuenta que algunos backends no proporcionan el correo electrónico del usuario por defecto, tienes que solicitarlo explícitamente, de lo contrario Weblate no será capaz de acreditar adecuadamente las contribuciones de los usuarios.

Consejo: La mayoría de los backends de autenticación requieren HTTPS. Una vez que el HTTPS esté habilitado en tu servidor web, por favor configura Weblate para que lo informe correctamente usando `ENABLE_HTTPS`, o mediante `WEBLATE_ENABLE_HTTPS` en el contenedor Docker.

Ver también:

[Dorsal de Python Social Auth](#)

Autenticación por OpenID

Para servicios basados en OpenID basta con activarlos. En esta sección se describe cómo activar la autenticación por OpenID de OpenSUSE, Fedora y Ubuntu:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.suse.OpenSUSEOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Ver también:

[OpenID](#)

Autenticación por GitHub

Tienes que registrar una aplicación OAuth en GitHub y luego decirle a Weblate todos sus secretos:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

GitHub debe estar configurado para tener una callback a la URL como `https://example.com/accounts/complete/github/`.

There are similar authentication backends for GitHub for Organizations and GitHub for Teams. Their settings are named `SOCIAL_AUTH_GITHUB_ORG_*` and `SOCIAL_AUTH_GITHUB_TEAM_*`, and they require additional setting of the scope - `SOCIAL_AUTH_GITHUB_ORG_NAME` or `SOCIAL_AUTH_GITHUB_TEAM_ID`. Their callback URLs are `https://example.com/accounts/complete/github-org/` and `https://example.com/accounts/complete/github-teams/`.

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver [Set correct site domain](#).

Ver también:

GitHub

Autenticación por Bitbucket

Tienes que registrar una aplicación en Bitbucket y luego decirle a Weblate todos sus secretos:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver [Set correct site domain](#).

Ver también:

Bitbucket

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+.

La URL de redirección es `https://WEBLATE_SERVER/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver [Set correct site domain](#).

Ver también:

[Google](#)

OAuth 2 de Facebook

Como es habitual con los servicios OAuth 2, tienes que registrar tu aplicación en Facebook. Una vez hecho esto, puedes configurar Weblate para utilizarla:

La URL de redirección es `https://WEBLATE_SERVER/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver [Set correct site domain](#).

Ver también:

[Facebook](#)

OAuth 2 de GitLab

Para utilizar GitLab OAuth 2, es necesario registrar una aplicación en <<https://gitlab.com/profile/applications>>.

La URL de redirección es `https://WEBLATE_SERVER/accounts/complete/gitlab/` y asegúrate de marcar el ámbito `read_user`.

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

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

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

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver [Set correct site domain](#).

Ver también:

[GitLab](#)

Active Directory de Microsoft Azure

Puede configurarse Weblate para utilizar inquilinos comunes o específicos para la autenticación.

La URL redireccionada es `https://WEBLATE_SERVER/accounts/complete/azuread-oauth2/` para las actividades comunes y `https://WEBLATE_SERVER/accounts/complete/azuread-tenant-oauth2/` para la autenticación específica del usuario.

```
# 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",
)
```

(continué en la próxima página)

(proviene de la página anterior)

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

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver [Set correct site domain](#).

Ver también:

Microsoft Azure Active Directory

Slack

Para utilizar Slack OAuth 2, necesitas registrar una aplicación en <https://api.slack.com/apps>.

La URL redireccionada es `https://WEBLATE_SERVER/accounts/complete/slack/`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.slack.SlackOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SLACK_KEY = ""
SOCIAL_AUTH_SLACK_SECRET = ""
```

Nota: La URL callback proporcionada por Weblate durante la autenticación incluye el dominio configurado. En caso de que se produzcan errores sobre la falta de coincidencia de la URL, es posible que desees arreglar esto, ver [Set correct site domain](#).

Ver también:

Slack

Anulación de los nombres e iconos de los métodos de autenticación

Puedes anular el nombre y el icono del método de autenticación utilizando la configuración como `SOCIAL_AUTH_<NAME>_IMAGE` y `SOCIAL_AUTH_<NAME>_TITLE`. Por ejemplo, el nombre de anulación para Auth0 se vería como:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Desactivar la autenticación por contraseña

La autenticación por correo electrónico y contraseña puede desactivarse eliminando `social_core.backends.email.EmailAuth` desde `AUTHENTICATION_BACKENDS`. Mantén siempre `weblate.accounts.auth.WeblateUserBackend`, es necesaria para la funcionalidad principal de Weblate.

Disabling e-mail authentication will disable all e-mail related functionality – user invitation or password reset feature.

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

Por ejemplo, la autenticación utilizando sólo el proveedor Open ID de openSUSE se puede lograr con lo siguiente:

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

2.5.4 Autenticación por contraseña

Por defecto `settings.py` viene con un set razonable de `AUTH_PASSWORD_VALIDATORS`:

- Las contraseñas no deben asemejarse demasiado a otros datos personales suyos.
- Las contraseñas deben tener por lo menos 10 caracteres de longitud.
- Las contraseñas no pueden figurar entre las más comúnmente utilizadas.
- Las contraseñas no pueden consistir enteramente de números.
- Las contraseñas no pueden consistir de un único carácter o de solo espacios.
- Las contraseñas no pueden coincidir con alguna que haya utilizado en el pasado.

Puede personalizar esta configuración para que se ajuste a su normativa de contraseñas.

Además, puedes instalar `django-zxcvbn-password` que ofrece estimaciones bastante realistas de la dificultad de las contraseñas y permite rechazar las contraseñas por debajo de un determinado umbral.

2.5.5 Autenticación por SAML

Nuevo en la versión 4.1.1.

Por favor, sigue las instrucciones de Python Social Auth para la configuración. Diferencias notables:

- Weblate soporta un único IDP que se debe llamar `weblate` en `SOCIAL_AUTH_SAML_ENABLED_IDPS`.
- La URL de los metadatos XML de SAML es `/accounts/metadata/saml/`.
- Los siguientes ajustes se rellenan automáticamente: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Ejemplo de configuración:

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

(continué en la próxima página)

(proviene de la página anterior)

```
# Social auth backends setup
SOCIAL_AUTH_SAML_SP_ENTITY_ID = f"https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbnl+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}
```

La configuración por defecto extrae los detalles del usuario de los siguientes atributos, configura tu IDP para proporcionarlos:

Atributo	Referencia de URI de SAML
Nombre completo	urn:oid:2.5.4.3
Nombre	urn:oid:2.5.4.42
Apellidos	urn:oid:2.5.4.4
Correo electrónico	urn:oid:0.9.2342.19200300.100.1.3
Nombre de usuario	urn:oid:0.9.2342.19200300.100.1.1

Consejo: El ejemplo anterior y la imagen Docker definen un IDP etiquetado: `weblate`. Es posible que tengas que configurar esta cadena como *Relay* en tu IDP.

Ver también:

Configuring SAML in Docker, SAML

2.5.6 Autenticación LDAP

La autenticación LDAP puede lograrse mejor utilizando el paquete *django-auth-ldap*. Puedes instalarlo por los medios habituales:

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

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

Consejo: Este paquete está incluido en el contenedor Docker, véase *Instalar con Docker*.

Nota: Hay algunas incompatibilidades en el módulo LDAP 3.1.0 de Python, que podrían impedirte utilizar esa versión. Si se produce el error `AttributeError: "module" object has no attribute "_trace_level"`, podría ayudar la actualización de python-ldap a la versión 3.0.0.

Una vez que tengas el paquete instalado, lo puedes enlazar a la autenticación de Django:

```
# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"

# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'

# List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
    # Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
    # 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
    "email": "mail",
}

# Hide the registration form
REGISTRATION_OPEN = False
```

Nota: Debes eliminar `'social_core.backends.email.EmailAuth'` de la configuración `AUTHENTICATION_BACKENDS`, de lo contrario los usuarios podrán establecer su contraseña en Weblate, y autenticarse usando eso. Sigue siendo necesario mantener `'weblate.accounts.auth.WeblateUserBackend'` para hacer los permisos y facilitar los usuarios anónimos. También te permitirá iniciar sesión con una cuenta de administrador local, si la has creado (por ejemplo, usando *createadmin*).

Uso de la contraseña de enlace

Si no puedes usar el enlace directo para la autenticación, tendrás que utilizar la búsqueda y proporcionar un usuario para el enlace de la búsqueda. Por ejemplo:

```
import ldap
from django_auth_ldap.config import LDAPSearch

AUTH_LDAP_BIND_DN = ""
AUTH_LDAP_BIND_PASSWORD = ""
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "ou=users,dc=example,dc=com", ldap.SCOPE_SUBTREE, "(uid=%(user)s)"
)
```

Integración con Active Directory

```
import ldap
from django_auth_ldap.config import LDAPSearch, NestedActiveDirectoryGroupType

AUTH_LDAP_BIND_DN = "CN=ldap,CN=Users,DC=example,DC=com"
AUTH_LDAP_BIND_PASSWORD = "password"

# User and group search objects and types
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "CN=Users,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s)"
)

# Make selected group a superuser in Weblate
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "OU=Groups,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True
```

Ver también:

[Django Authentication Using LDAP, Authentication](#)

2.5.7 Autenticación CAS

Se puede implantar una autenticación CAS al utilizar un paquete como *django-cas-ng*.

El primer paso consiste en revelar el campo Correo electrónico del usuario mediante CAS. Esto debe configurarse en el propio servidor CAS, y necesitará ejecutar al menos la versión 2 de CAS, ya que CAS v1 no admite atributos.

El segundo paso será actualizar Weblate para que utilice el servidor y los atributos de CAS.

Para instalar *django-cas-ng*:

```
pip install django-cas-ng
```

Una vez que haya instalado el paquete, puede conectarlo con el sistema de autenticación de Django; para ello, modifique el archivo `settings.py`:

```
# Add CAS backed, keep the Django one if you want to be able to sign in
# even without LDAP for the admin account
AUTHENTICATION_BACKENDS = (
    "django_cas_ng.backends.CASBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# CAS server address
CAS_SERVER_URL = "https://cas.example.net/cas/"

# Add django_cas_ng somewhere in the list of INSTALLED_APPS
INSTALLED_APPS = (... , "django_cas_ng")
```

Finalmente, se puede utilizar una señal para vincular el campo Correo electrónico y el objeto de usuario. Para que esto funcione, debe importar la señal del paquete *django-cas-ng* y conectar su código con esta señal. Realizar esto en el archivo de configuración puede causar problemas, por lo cual se recomienda ponerlo:

- En el método `django.apps.AppConfig.ready()` de la configuración de su aplicación
- En el archivo `urls.py` del proyecto (cuando no existan modelos)

```
from django_cas_ng.signals import cas_user_authenticated
from django.dispatch import receiver

@receiver(cas_user_authenticated)
def update_user_email_address(sender, user=None, attributes=None, **kwargs):
    # If your CAS server does not always include the email attribute
    # you can wrap the next two lines of code in a try/catch block.
    user.email = attributes["email"]
    user.save()
```

Ver también:

[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: Hasta Weblate 3.0, el sistema de privilegios se basaba únicamente en el de Django, pero ahora se ha construido específicamente para Weblate. Si utiliza una versión antigua, consulte la documentación que corresponda a esa versión.

2.6.1 Control de acceso simple

Si no está administrando toda la instalación de Weblate y solo tiene acceso para administrar ciertos proyectos (como en *Hosted Weblate* <<https://hosted.weblate.org/>> _), sus opciones de administración de control de acceso se limitan a las siguientes configuraciones. Si no necesita ninguna configuración compleja, será suficiente para ti.

Control de acceso al proyecto

Nota: Esta función no está disponible para los proyectos que ejecutan el plan Libre en Hosted Weblate.

Puede limitar el acceso del usuario a proyectos individuales seleccionando una configuración diferente: guíalel: *Control de acceso*. Las opciones disponibles son:

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

Protegido Visible públicamente y traducible solo por usuarios seleccionados.

Privado Solo visible para, y traducible por, usuarios seleccionados.

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 screenshot shows the Weblate web interface. At the top is a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a breadcrumb 'WeblateOrg / Settings'. The 'Access' tab is active, with other tabs being 'Basic', 'Workflow', and 'Components'. Under 'Access control', there are four radio button options: 'Public' (unselected), 'Protected' (selected), 'Private' (unselected), and 'Custom' (unselected). Each option has a brief description of its permissions. A message at the bottom indicates a lack of permission to change the access control, with a link to 'Check your billing status'. A 'Save' button is at the bottom left. The footer contains 'Powered by Weblate 4.12' and various links like 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

El valor predeterminado se puede cambiar mediante :setting: *DEFAULT_ACCESS_CONTROL*.

Nota: Incluso para los proyectos *Privados*, alguna información sobre su proyecto será expuesta: las estadísticas y el resumen de idiomas para toda la instancia incluirán los recuentos de todos los proyectos a pesar de la configuración del control de acceso. El nombre de su proyecto y otra información no pueden ser revelados a través de esto.

Nota: El administrador de la instancia de Weblate puede redefinir el conjunto de permisos disponibles por defecto para los usuarios en los proyectos *Public*, *Protected*, y *Private* utilizando *custom settings*.

Advertencia: Al activar el control de acceso *Personalizado*, Weblate eliminará todos los *grupos especiales* que haya creado para un proyecto seleccionado. Si hace esto sin permiso de administrador para toda la instancia de Weblate, perderá instantáneamente su acceso para gestionar el proyecto.

Ver también:

Control de acceso

Gestionar el control de acceso por proyecto

Los usuarios con el privilegio *Manage project access* (ver *Lista de privilegios y roles incorporados*) pueden gestionar los usuarios en los proyectos mediante la adición de ellos a los equipos. La colección inicial de equipos es proporcionada por Weblate, pero se pueden definir otros adicionales que proporcionan idiomas y asignarles roles de acceso designados (ver privilegios).

Los siguientes equipos se crean automáticamente para cada proyecto:

Para proyectos *Públicos*, *Protegidos* y *Privados*:

Administración Cuenta con todos los permisos disponibles en el proyecto.

Revisión (sólo si *review workflow* está activado) Puede aprobar traducciones durante la revisión.

Para proyectos *protegidos* y *privados* únicamente:

Traducir Puede traducir el proyecto y cargar traducciones realizadas sin conexión.

Orígenes Puede editar las cadenas de origen (si se permite en la *configuración del proyecto*) y la información de la cadena de origen.

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

Glosario Puede gestionar el glosario (añadir o quitar entradas, o bien cargarlas).

Memoria Puede gestionar la memoria de traducción.

Capturas de pantalla Puede gestionar las capturas de pantalla (añadirlas o eliminarlas, así como asociarlas a cadenas de origen).

Traducción automática Puede usar traducción automática.

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

Facturación Puede acceder a la información y la configuración de facturación (vea *Facturación*).

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 breadcrumb trail: WeblateOrg / Access control. The main content area has three tabs: Users (active), Teams, and API access. Below the tabs is a 'Users' table with columns: Username, Full name, E-mail, Last sign in, and Teams. The table contains one user: 'testuser' with full name 'Weblate Test', email 'weblate@example.org', and last sign in '21 seconds ago'. There is a 'Translate' button and an edit icon next to the user. Below the table is a note: 'Once all its permissions are removed, the user will be removed from the project.' To the left of the table are two forms: 'Add a user' and 'Block user'. To the right is an 'Invite new user' form. The footer contains links: Powered by Weblate 4.12, About Weblate, Legal, Contact, Documentation, and Donate to Weblate.

Username	Full name	E-mail	Last sign in	Teams
testuser	Weblate Test	weblate@example.org	21 seconds ago	Translate

Once all its permissions are removed, the user will be removed from the project.

Add a user

User to add

Please type in an existing Weblate account name or e-mail address.

[Add](#)

Block user

User to block

Please type in an existing Weblate account name or e-mail address.

Block duration

Block user until I unblock them

[Block](#)

Invite new user

E-mail

Username

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

Full name

[Invite](#)

Powered by Weblate 4.12 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Estas funciones están disponibles en la página *Control de acceso*, a la que se puede acceder desde el menú del proyecto *Gestión ↓ Usuarios*.

Invitación a usuarios nuevos

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* team) would be sufficient.

Consejo: Si el usuario invitado pierde la validación de la invitación, pueden establecer su contraseña utilizando la dirección de correo electrónico invitado en el formulario de restablecimiento de contraseña, ya que la cuenta ya está creada.

Nuevo en la versión 3.11: Es posible reenviar el correo electrónico para las invitaciones de los usuarios (invalidando cualquier invitación enviada anteriormente).

El mismo tipo de invitaciones están disponibles en todo el sitio desde la *management interface* en la pestaña *Users*.

Bloquear usuarios

Nuevo en la versión 4.7.

En caso de que algunos usuarios se comporten mal en tu proyecto, tienes la opción de bloquearlos para que no contribuyan. El usuario bloqueado podrá seguir viendo el proyecto si tiene permisos para ello, pero no podrá contribuir.

Gestión de permisos por proyecto

Puede definir sus proyectos como *Protegido* o *Privado* y *gestionar los usuarios* por proyecto en la interfaz de usuario de Weblate.

Por defecto, esto evita que Weblate conceda el acceso proporcionado por *Users* y *Viewers default groups* debido a la propia configuración de estos grupos. Esto no impide que se concedan permisos a esos proyectos en todo el sitio modificando los grupos por defecto, creando uno nuevo o creando configuraciones personalizadas adicionales para componentes individuales como se describe en *Control de acceso personalizado* más abajo.

Uno de los principales beneficios de administrar permisos a través de la interfaz de usuario de Weblate es que puedes delegarlo a otros usuarios sin darles el privilegio de superusuario. Para ello, agréguelos al equipo de *Administración* del proyecto.

2.6.2 Control de acceso personalizado

Nota: Esta función no está disponible para los proyectos que ejecutan el plan Libre en Hosted Weblate.

El sistema de permisos está basado en grupos y roles, donde los roles definen un conjunto de permisos, y los grupos los vinculan a los usuarios y a las traducciones, véase *Usuarios, roles, grupos y permisos* para más detalles.

Las características más potentes del sistema de control de acceso de Weblate por ahora sólo están disponibles a través de la interfaz de administración de *Django admin interface*. Puedes utilizarlo para gestionar los permisos de cualquier proyecto. No tienes que cambiarlo necesariamente a *Custom access control* para usarlo. Sin embargo, debes tener privilegios de superusuario para poder utilizarlo.

Si no estás interesado en los detalles de la implementación, y sólo quieres crear una configuración muy simple basada en los valores predeterminados, o no tienes un acceso a todo el sitio de la instalación de Weblate (como en *Hosted Weblate*), por favor visita la sección *Control de acceso simple*.

Configuraciones comunes

Esta sección contiene un resumen de algunas configuraciones comunes que pueden interesarte.

Gestión de permisos para todo el sitio

Para gestionar los permisos de toda una instancia a la vez, añade usuarios a los grupos apropiados *default groups*:

- *Users* (esto se hace por defecto mediante la *automatic group assignment*).
- *Reviewers* (si estás utilizando *review workflow* con revisores dedicados).
- *Managers* (si quieres delegar la mayor parte de las operaciones de gestión a otra persona).

Deberías mantener todos los proyectos configurados como *Public* (ver *Control de acceso al proyecto*), de lo contrario los permisos de todo el sitio proporcionados por la pertenencia a los grupos *Users* y *Reviewers* no tendrán ningún efecto.

También puedes conceder algunos permisos adicionales de tu elección a los grupos por defecto. Por ejemplo, puedes querer dar un permiso para gestionar las capturas de pantalla a todos los *Users*.

También puedes definir nuevos grupos personalizados. Si quieres seguir gestionando tus permisos en todo el sitio para estos grupos, elige un valor apropiado para *Project selection* (e.g. *All projects* o *All public projects*).

Permisos personalizados para idiomas, componentes o proyectos

Puedes crear tus propios grupos dedicados para gestionar los permisos de distintos objetos como idiomas, componentes y proyectos. Aunque estos grupos sólo pueden conceder privilegios adicionales, no puedes revocar ningún permiso concedido por los grupos de todo el sitio o por proyecto añadiendo otro grupo personalizado.

Ejemplo:

Si quieres (por la razón que sea) permitir la traducción a un idioma específico (digamos *checo*) sólo a un conjunto cerrado de traductores fiables mientras mantienes las traducciones a otros idiomas públicos, tendrás que hacerlo:

1. Elimina el permiso para traducir *Czech* de todos los usuarios. En la configuración por defecto esto se puede hacer alterando el grupo *Users default group*.

Tabla 1: Grupo *Usuarios*

Selección de idioma	<i>Como se ha definido</i>
Idiomas	Todos menos <i>checo</i>

2. Añadir un grupo dedicado para los traductores de *checo*.

Tabla 2: Grupo *Traductores al checo*

Roles	<i>Usuario avanzado</i>
Selección del proyecto	<i>Todos los proyectos públicos</i>
Selección de idioma	<i>Como se ha definido</i>
Idiomas	<i>Checo</i>

3. Añade a este grupo los usuarios a los que quieras dar permisos.

Como puedes ver, la gestión de permisos de esta manera es poderosa, pero puede ser un trabajo bastante tedioso. No se puede delegar a otro usuario, a no ser que se concedan permisos de superusuario.

Usuarios, roles, grupos y permisos

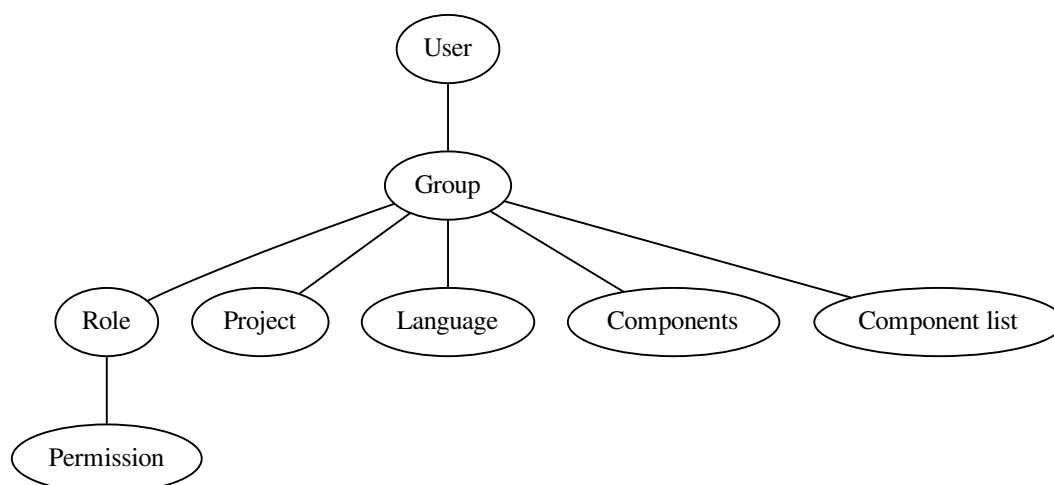
Los modelos de autenticación consisten de varios objetos:

Permiso Permiso individual definido por Weblate. Los permisos no se pueden asignar a los usuarios. Esto sólo puede hacerse mediante la asignación de roles.

Rol Un rol define un conjunto de permisos. Esto permite reutilizar este conjunto en varios lugares, facilitando la administración.

Usuario Un usuario puede ser miembro de varios grupos.

Grupo El grupo conecta roles, usuarios y objetos de autenticación (proyectos, idiomas y listas de componentes).



Nota: Un grupo puede no tener roles asignados, en ese caso se asume el acceso a navegar por el proyecto por parte de cualquiera (ver más abajo).

Acceso a navegar por un proyecto

Un usuario tiene que ser miembro de un grupo vinculado al proyecto, o a cualquier componente dentro de ese proyecto. Tener la membresía es suficiente, no se necesitan permisos específicos para navegar por el proyecto (esto se utiliza en el grupo por defecto *Viewers*, ver [Lista de grupos](#)).

Acceso para navegar a un componente

Un usuario puede acceder a los componentes sin restricciones una vez que pueda acceder al proyecto de los componentes (y tendrá todos los permisos que el usuario tenga concedidos para el proyecto). Con [Acceso restringido](#) activado, el acceso al componente requiere permisos explícitos para el componente (o una lista de componentes en la que esté el componente).

Ámbito de los grupos

El alcance del permiso asignado por los roles en los grupos se aplica mediante las siguientes reglas:

- Si el grupo especifica cualquier *Component list*, todos los permisos otorgados a los miembros de ese grupo se conceden para todos los componentes de las listas de componentes adjuntas al grupo, y se concede un acceso sin permisos adicionales para todos los proyectos en los que estén estos componentes. *Components* y *Projects* se ignoran.
- Si el grupo especifica cualquier *Components*, todos los permisos otorgados a los miembros de ese grupo se conceden para todos los componentes adscritos al grupo, y se concede un acceso sin permisos adicionales para todos los proyectos en los que están estos componentes. *Projects* se ignora.
- De lo contrario, si el grupo especifica cualquier *Projects*, ya sea enumerándolos directamente o teniendo *Projects selection* fijado en un valor como *All public projects*, todos esos permisos se aplican a todos los proyectos, lo que efectivamente otorga los mismos permisos para acceder a todos los proyectos *unrestricted components*.

- Las restricciones impuestas por la *Languages* de un grupo se aplican por separado, cuando se verifica si un usuario tiene acceso a realizar determinadas acciones. En concreto, se aplica sólo a las acciones directamente relacionadas con el proceso de traducción en sí, como revisar, guardar traducciones, añadir sugerencias, etc.

Consejo: Para automatizar la inclusión de todos los idiomas o proyectos, usa *Language selection* o *Project selection*.

Ejemplo:

Digamos que hay un proyecto `foo` con los componentes: `foo/bar` y `foo/baz` y el grupo siguiente:

Tabla 3: Grupo *Administradores-Revisores Españoles*

Roles	<i>Revisar cadenas, Gestionar repositorio</i>
Componentes	<code>foo/bar</code>
Idiomas	<i>Español</i>

Los miembros de ese grupo tendrán los siguientes permisos (asumiendo la configuración de roles por defecto):

- Acceso general (navegación) a todo el proyecto `foo` incluyendo los dos componentes que lo componen: `foo/bar` y `foo/baz`.
- Revisar las cadenas en la traducción al español de `foo/bar` (no en otra parte).
- Gestionar VCS para todo el repositorio `foo/bar`, por ejemplo, commit los cambios pendientes realizados por los traductores para todos los idiomas.

Asignaciones de grupo automático

En la parte inferior de la página de edición de *Group* en la interfaz de administración de *Django*, puedes especificar *Automatic group assignments*, que es una lista de expresiones regulares utilizadas para asignar automáticamente a los usuarios recién creados a un grupo en función de sus direcciones de correo electrónico. Esta asignación sólo se realiza al crear la cuenta.

El caso de uso más común para esta función es asignar todos los nuevos usuarios a algún grupo por defecto. Para ello, probablemente querrás mantener el valor por defecto (`^.*$`) en el campo de expresión regular. Otro caso de uso para esta opción podría ser dar algunos privilegios adicionales a los empleados de tu empresa por defecto. Suponiendo que todos ellos utilicen direcciones de correo electrónico corporativas en su dominio, esto puede lograrse con una expresión como `^.*@miempresa.com`.

Nota: La asignación automática de grupos a *Usuarios* y *Visores* siempre se recrea al actualizar de una versión de Weblate a otra. Si deseas desactivarla, fija la expresión regular a `^$` (que no coincidirá con nada).

Nota: Por el momento, no hay forma de añadir en masa usuarios ya existentes a algún grupo a través de la interfaz de usuario. Para ello, puedes recurrir a la API [REST](#).

Grupos y funciones por defecto

Tras la instalación, se crea un conjunto de grupos por defecto (ver [Lista de grupos](#)).

Estos roles y grupos se crean en el momento de la instalación. Los roles incorporados se mantienen siempre actualizados por la migración de la base de datos cuando se actualiza. En realidad no puedes cambiarlos, por favor define un nuevo rol si quieres definir tu propio conjunto de permisos.

Lista de privilegios y roles incorporados

Alcance	Permiso	Roles
Facturación (vea Facturación)	Ver datos de facturación	Administración, Facturación
Cambios	Descargar cambios	Administración
Comentarios	Enviar comentario	Administración, Editar fuente, Usuario avanzado
	Eliminar comentario	Administración
Componente	Editar configuración de los componentes	Administración
	Bloquear componente, impidiendo traducciones	Administración
Glosario	Añadir palabra al glosario	Administración, Gestionar glosario, Usuario avanzado
	Editar palabra del glosario	Administración, Gestionar glosario, Usuario avanzado
	Eliminar palabra del glosario	Administración, Gestionar glosario, Usuario avanzado
	Cargar entradas de glosario	Administración, Gestionar glosario, Usuario avanzado
Sugerencias automáticas	Utilizar sugerencias automáticas	Administración, Editar fuente, Usuario avanzado
Memoria de traducción	Editar la memoria de traducción	Administración, Gestión de la memoria de traducción
	Eliminar memoria de traducción	Administración, Gestión de la memoria de traducción
Proyectos	Editar configuración del proyecto	Administración
	Gestionar acceso al proyecto	Administración
Informes	Descargar informes	Administración
Capturas de pantalla	Añadir captura de pantalla	Administración, Gestión de capturas de pantalla
	Editar captura de pantalla	Administración, Gestión de capturas de pantalla
	Eliminar captura de pantalla	Administración, Gestión de capturas de pantalla
Cadenas de origen	Editar información adicional de cadena	Administración, Editar código
Cadenas	Añadir nueva cadena	Administración
	Quitar una cadena	Administración
	Descartar comprobación fallida	Administración, Editar fuente, Usuario avanzado
	Editar cadenas	Administración, Editar fuente, Usuario avanzado
	Revisar cadenas	Administración, Revisar cadenas
	Editar cadena cuando son obligatorias las sugerencias	Administración, Revisar cadenas
	Editar cadenas de origen	Administración, Editar origen, Usuario avanzado
Sugerencias	Aceptar sugerencia	Administración, Editar fuente, Usuario avanzado
	Añadir sugerencia	Administración, Editar fuente, Añadir sugerencia
	Eliminar sugerencia	Administración, Usuario avanzado
	Votar sugerencia	Administración, Editar fuente, Usuario avanzado
Traducciones	Añadir idioma para traducción	Administración, Usuario avanzado, Administración de idiomas
	Realizar una traducción automática	Administración, Gestión de idiomas
	Eliminar traducción existente	Administración, Gestión de idiomas
	Descargar archivo de traducción	Administración, Edit source, Access repository
	Añadir varios idiomas para traducción	Administración, Gestión de idiomas
Cargas	Definir autor/a de la traducción cargada	Administración
	Sobrescribir cadenas existentes al subir	Administración, Editar fuente, Usuario avanzado
	Cargar traducciones	Administración, Editar fuente, Usuario avanzado
Sistema de control de versiones	Acceso al repositorio interno	Administración, Access repository, Power user
	Enviar cambios al repositorio interno	Administración, Manage repository
	Enviar cambios del repositorio interno	Administración, Manage repository
	Deshacer los cambios del repositorio interno	Administración, Manage repository
	Ver la ubicación del repositorio originario	Administración, Access repository, Power user

Tabla 4 – proviene de la página anterior

Alcance	Permiso	Roles
	Actualizar el repositorio interno	<i>Administration, Manage repository</i>
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: Los privilegios para todo el sitio no se conceden a ningún rol por defecto. Estos son poderosos y bastante cercanos al estatus de superusuario. La mayoría de ellos afectan a todos los proyectos en tu instalación de Weblate.

Lista de grupos

Los siguientes grupos se crean en el momento de la instalación (o después de ejecutar *setupgroups*) y eres libre de modificarlos. Sin embargo, la migración los volverá a crear si los eliminas o les cambias el nombre.

Invitados Define los permisos que tendrán los usuarios no autenticados.

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

Puedes eliminar los roles de este grupo para limitar los permisos de los usuarios no autenticados.

Roles por defecto: *Añadir sugerencia, Acceder al repositorio*

Lectores Este rol asegura la visibilidad de los proyectos públicos para todos los usuarios. Por defecto, todos los usuarios son miembros de este grupo.

Por defecto, asignación automática de grupo hace que, cuando se unen, todas las cuentas nuevas sean miembros de este grupo.

Roles por defecto: ninguno

Usuario/as Grupo predeterminado para todos los usuarios.

Por defecto, asignación automática de grupo hace que, cuando se unen, todas las cuentas nuevas sean miembros de este grupo.

Roles por defecto: *Usuario/a avanzado/a*

Revisores Grupo de revisores (ver *Flujos de trabajo de traducción*).

Roles por defecto: *Cadenas de revisión*.

Supervisores Grupo para administradores.

Roles por defecto: *Administración*

Advertencia: Nunca elimines grupos y usuarios predefinidos de Weblate, ya que esto puede provocar problemas inesperados. Si no los utilizas, puedes eliminar todos sus privilegios.

2.6.3 Restricciones de acceso adicionales

Si deseas utilizar tu instalación de Weblate de una manera menos pública, es decir, permitir nuevos usuarios sólo por invitación, puedes hacerlo configurando Weblate de tal manera que sólo los usuarios conocidos tengan acceso a él. Para ello, puedes fijar `REGISTRATION_OPEN` a `False` para evitar el registro de nuevos usuarios, y fijar `REQUIRE_LOGIN` a `/*` para requerir el inicio de sesión para acceder a todas las páginas del sitio. Esta es básicamente la forma de bloquear tu instalación de Weblate.

Consejo: Puedes usar la función *invitaciones* para añadir nuevos usuarios.

2.7 Proyectos de traducción

2.7.1 Organización de traducción

Weblate organiza el contenido traducible del sistema de control de versiones del proyecto o los componentes en una estructura semejante a un árbol.

- The bottom level object is *Configuración de proyectos*, which should hold all translations belonging together (for example translation of an application in several versions and/or accompanying documentation).
- On the level above, *Configuración de componentes*, which is actually the component to translate, you define the VCS repository to use, and the mask of files to translate.
- Above *Configuración de componentes* there are individual translations, handled automatically by Weblate as translation files (which match *Máscara de archivos* defined in *Configuración de componentes*) appear in the VCS repository.

Weblate supports a wide range of translation formats (both bilingual and monolingual ones) supported by Translate Toolkit, see *Formatos de archivo admitidos*.

Nota: You can share cloned VCS repositories using *URL internos de Weblate*. Using this feature is highly recommended when you have many components sharing the same VCS. It improves performance and decreases required disk space.


2.7.2 Añadir proyectos y componentes de traducción

Distinto en la versión 3.2: An interface for adding projects and components is included, and you no longer have to use *La interfaz administrativa de Django*.

Distinto en la versión 3.4: The process of adding components is now multi staged, with automated discovery of most parameters.

Based on your permissions, new translation projects and components can be created. It is always permitted for users with the *Add new projects* permission, and if your instance uses billing (e.g. like <https://hosted.weblate.org/> see *Facturación*), you can also create those based on your plans allowance from the user account that manages billing.

Puede ver su plan de facturación actual en una página separada:

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

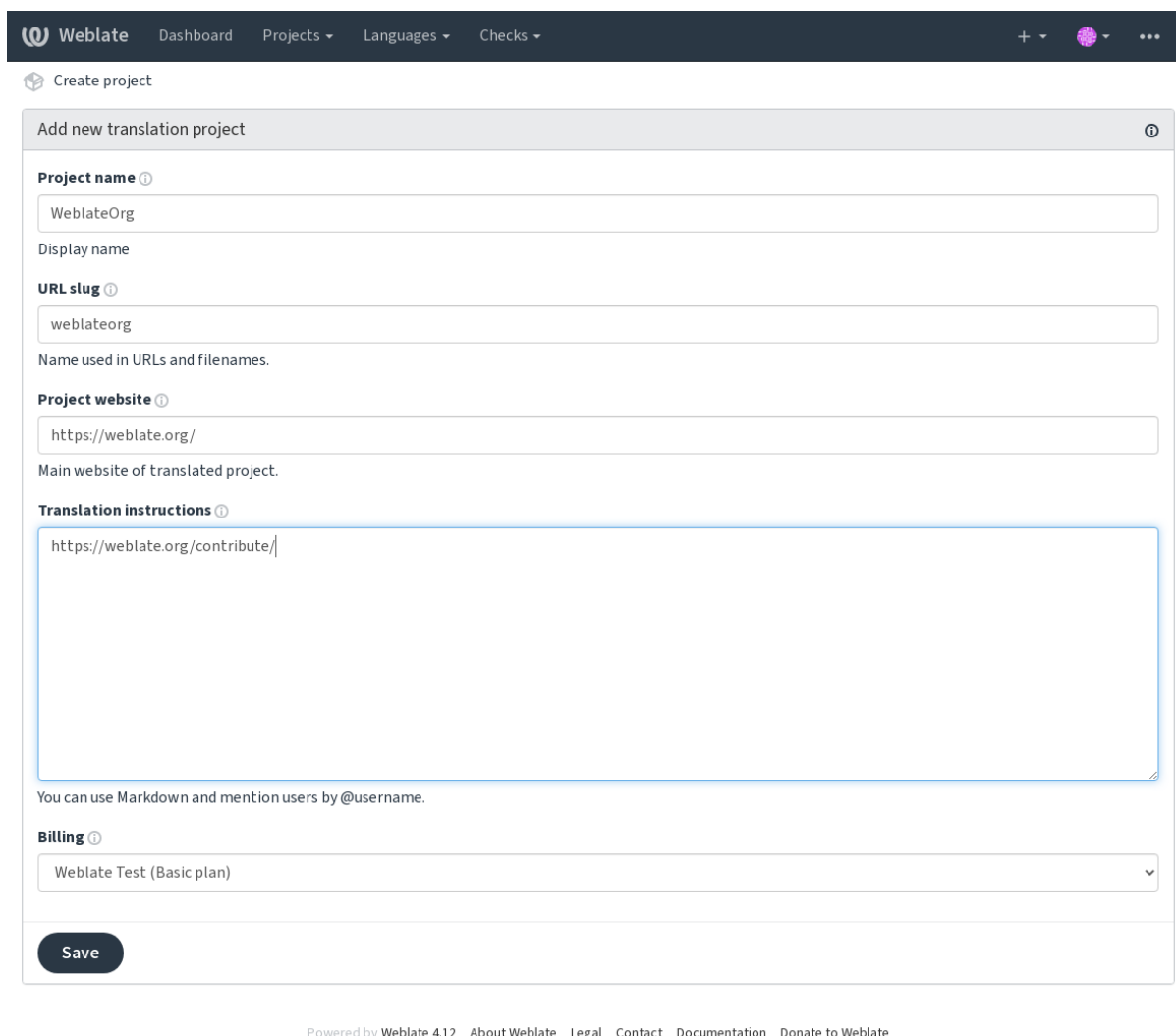
[Your profile](#) / [Billing](#)

Billing plan		ⓘ
Current plan	Basic plan (Active)	
Monthly price	19 EUR	
Yearly price	199 EUR	
Strings limit	Used 0	<div></div>
Languages limit	Used 0	<div></div>
Last invoice	2022-04-19 - 2022-04-21	
Projects limit	Used 0 of 1	<div></div>
Projects	No projects currently assigned! Add new translation project	
Terminate billing plan		

Invoices		
Invoice period	Invoice amount	Download invoice
04/19/2022 - 04/21/2022	19.0 EUR	Not available

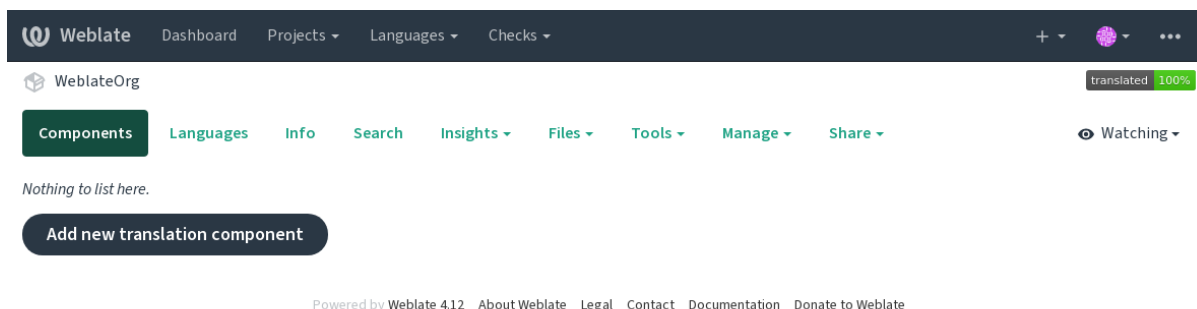
Powered by Weblate 4.12 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

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:



The screenshot shows the 'Add new translation project' form in the Weblate interface. The form is titled 'Add new translation project' and includes several input fields and a text area. The fields are: 'Project name' (containing 'WeblateOrg'), 'URL slug' (containing 'weblateorg'), 'Project website' (containing 'https://weblate.org/'), and 'Translation instructions' (containing 'https://weblate.org/contribute/'). There is also a 'Billing' dropdown menu set to 'Weblate Test (Basic plan)'. A 'Save' button is at the bottom. The footer of the page reads 'Powered by Weblate 4.12 About Weblate Legal Contact Documentation Donate to Weblate'.

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



The screenshot shows the project page for 'WeblateOrg' in the Weblate interface. The page has a dark header with the Weblate logo and navigation links: 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below the header, there is a 'WeblateOrg' section with a 'translated 100%' indicator. A 'Components' button is highlighted, and there are links for 'Languages', 'Info', 'Search', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. A 'Watching' button is also present. Below this, there is a message 'Nothing to list here.' and an 'Add new translation component' button. The footer of the page reads 'Powered by Weblate 4.12 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 Upload single document or translation file and translate that.

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:

Create component

From version control Upload translations files Translate document Start from scratch

Create a new translation component from remote version control system repository.

Component name ⓘ

Language names

Display name

URL slug ⓘ

language-names

Name used in URLs and filenames.

☐ Use as a glossary ⓘ

Project ⓘ

WeblateOrg

Source language ⓘ

English

Language used for source strings in all components

Version control system ⓘ

Git

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository ⓘ

https://github.com/WeblateOrg/demo.git

URL of a repository, use weblate://project/component to share it with other component.

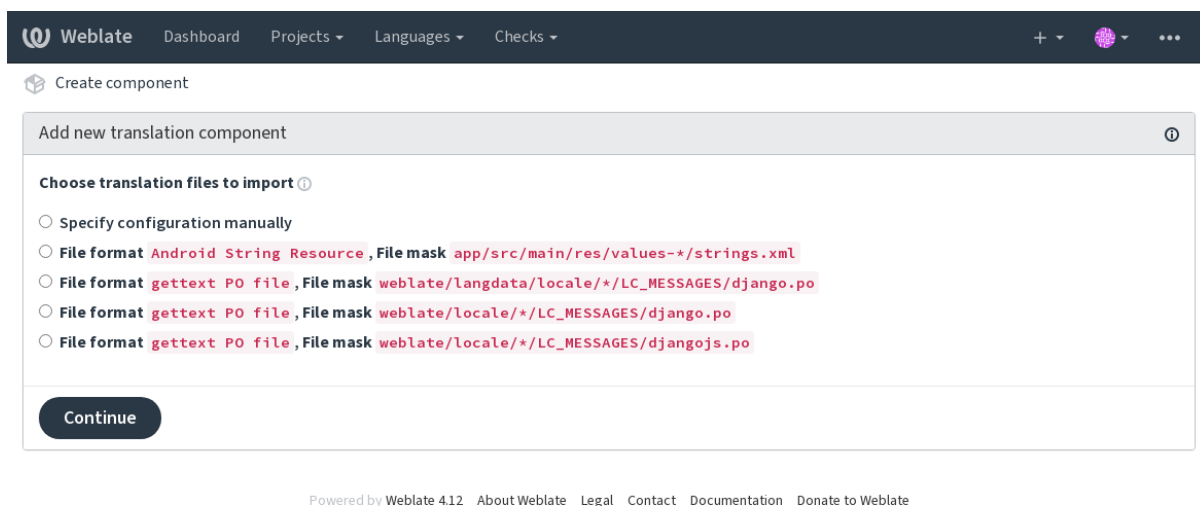
Repository branch ⓘ

Repository branch to translate

Continue

Powered by Weblate 4.12 About Weblate Legal Contact Documentation Donate to Weblate

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

 Weblate Dashboard Projects Languages Checks

Create component

Add new translation component

Choose translation files to import

☐ Specify configuration manually

☐ File format `Android String Resource`, File mask `app/src/main/res/values-*/strings.xml`

☐ File format `gettext PO file`, File mask `weblate/langdata/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/django.po`

☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/djangojs.po`

Continue

Powered by Weblate 4.12 About Weblate Legal Contact Documentation Donate to Weblate

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

Weblate

Dashboard

Projects

Languages

Checks

+

▼

●

...

Create component

Add new translation component

Project

WeblateOrg

Component name

Language names

Display name

URL slug

language-names

Name used in URLs and filenames.

Version control system

Git

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository

https://github.com/WeblateOrg/demo.git

URL of a repository, use weblate://project/component to share it with other component.

Repository branch

Repository branch to translate

Repository push URL

URL of a push repository, pushing is turned off if empty.

Push branch

Branch for pushing changes, leave empty to use repository branch

Repository browser

https://github.com/WeblateOrg/demo/blob/{{branch}}/{{filename}}#L{{line}}

Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. You might want to strip leading directory by using {{filename|parentdir}}.

File format

gettext PO file

File mask

app/src/main/res/values-*/strings.xmlweblate/langdata/locale/*/LC_MESSAGES/django.po

Path of files to translate relative to repository root, use * instead of language code, for example: po/* or locale/*/LC_MESSAGES/django.po.

Monolingual base language file

app/src/main/res/values/strings.xml

Filename of translation base file, containing all strings and their source; it is recommended for monolingual translation formats.

☒ Edit base file

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Template for new translations

weblate/langdata/locale/django.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Translation license

GNU General Public License v3.0 or later

Adding new translation

Create new language file

How to handle requests for creating new translations.

Language code style

Default based on the file format

Customize language code used to generate the filename for translations created by Weblate.

Language filter

^(cs|he|hu)\$

Regular expression used to filter translation files when scanning for file mask.

Source language

English

Language used for source strings in all components

☐ Use as a glossary

You will be able to edit more options in the component settings after creating it.

Save

Powered by Weblate 4.12 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

Text describing localization process in the project, and any other information useful for translators. Markdown can be used for text formatting or inserting links.

Definir cabecera «Language-Team»

Whether Weblate should manage the Language-Team header (this is a *gettext de GNU* only feature right now).

Utilizar memoria de traducción compartida

Whether to use shared translation memory, see *Memoria de traducción compartida* for more details.

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*, *Definiciones de idioma*

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:

Procesamiento de códigos de idioma

2.7.4 Configuración de componentes

A component is a grouping of something for translation. You enter a VCS repository location and file mask for which files you want translated, and Weblate automatically fetches from this VCS, and finds all matching translatable files.

Ver también:

[/devel/integration](#)

You can find some examples of typical configurations in the *Formatos de archivo admitidos*.

Nota: It is recommended to keep translation components to a reasonable size - split the translation by anything that makes sense in your case (individual apps or add-ons, book chapters or websites).

Weblate 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.

For linked repositories, this is not used and setting from linked component applies.

Ver también:

See *Accessing repositories* for more details on how to specify a repository URL and *Enviar cambios efectuados en Weblate* for more details on pushing changes from Weblate.

Explorador del repositorio

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use *Marcación de plantilla*.

For example on GitHub, use something like: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

In case your paths are relative to different folder (path contains `..`), you might want to strip leading directory by `parentdir` filter (see *Marcación de plantilla*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

URL de repositorio exportado

URL where changes made by Weblate are exported. This is important when *Regionalización continua* is not used, or when there is a need to manually merge changes. You can use *Git exporter* to automate this for Git repositories.

Rama del repositorio

Which branch to checkout from the VCS, and where to look for translations.

For linked repositories, this is not used and setting from linked component applies.

Rama a la que enviar

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

For linked repositories, this is not used and setting from linked component applies.

Nota: This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

Ver también:

Enviar cambios efectuados en Weblate

Máscara de archivos

Mask of files to translate, including path. It should include one «*» replacing language code (see *Definiciones de idioma* for info on how this is processed). In case your repository contains more than one translation file (e.g. more gettext domains), you need to create a component for each of them.

For example `po/* .po` or `locale/*/LC_MESSAGES/django.po`.

In case your filename contains special characters such as `[,]`, these need to be escaped as `[[]` or `[]]`.

Ver también:

Formatos bilingües y monolingües, What does mean «There are more files for the single language (en)»?

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 empty file by default. Use this in case you want to have all strings present with empty value when creating new translation.

Ver también:

adding-translation, *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`.

Ver también:

Mantener iguales las traducciones entre los componentes

Activar sugerencias

Whether translation suggestions are accepted for this component.

Votar sugerencias

Turns on vote casting for suggestions, see *Votar sugerencias*.

Aceptar sugerencias automáticamente

Automatically accept voted suggestions, see *Votar sugerencias*.

Indicadores de traducción

Customization of quality checks and other Weblate behavior, see *Personalizar el comportamiento mediante indicadores*.

Comprobaciones obligatorias

List of checks which can not be ignored, see *Forzar comprobaciones*.

Nota: Enforcing the check does not automatically enable it, you still should enabled it using *Personalizar el comportamiento mediante indicadores* in *Indicadores de traducción* or *Información adicional sobre las cadenas de origen*.

Licencia de la traducción

License of the translation (does not need to be the same as the source code license).

Acuerdo de contribuidor

Acuerdo de usuario que debe aprobarse antes de que un usuario pueda traducir este componente.

Adición de traducciones nuevas

How to handle requests for creation of new languages. Available options:

Contactar a responsables User can select desired language and the project maintainers will receive a notification about this. It is up to them to add (or not) the language to the repository.

Apuntar al URL con instrucciones de traducción User is presented a link to page which describes process of starting new translations. Use this in case more formal process is desired (for example forming a team of people before starting actual translation).

Crear archivo de idioma nuevo User can select language and Weblate automatically creates the file for it and translation can begin.

Desactivar adición de traducciones nuevas There will be no option for user to start new translation.

Consejo: The project admins can add new translations even if it is disabled here when it is possible (either *Plantilla para traducciones nuevas* or the file format supports starting from an empty file).

Ver también:

adding-translation, *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, Procesamiento de códigos de idioma

Estilo de fusión

You can configure how updates from the upstream repository are handled. The actual implementation depends on VCS, see *Integración de control de versiones*.

Cambiar base («rebase») Rebases Weblate commits on top of upstream repository on update. This provides clean history without extra merge commits.

Rebasing can cause you trouble in case of complicated merges, so carefully consider whether or not you want to enable them.

You might need to enable force pushing by choosing *Git con envío forzado* as *Sistema de control de versiones*, especially when pushing to a different branch.

Fusionar Upstream repository changes are merged into Weblate one. This setting utilizes fast-forward when possible. This is the safest way, but might produce a lot of merge commits.

Merge without fast-forward Upstream repository changes are merged into Weblate one with doing a merge commit every time (even when fast-forward would be possible). Every Weblate change will appear as a merge commit in Weblate repository.

Default value can be changed by *DEFAULT_MERGE_STYLE*.

Commit, add, delete, merge, add-on, and merge request messages

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*, *DEFAULT_PULL_MESSAGE*.

Enviar al consignar

Whether committed changes should be automatically pushed to the upstream repository. When enabled, the push is initiated once Weblate commits changes to its underlying repository (see *Consignas diferidas*). To actually enable pushing *Repository push URL* has to be configured as well.

Antigüedad de cambios por consignar

Sets how old (in hours) changes have to be before they are committed by background task or the *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 file mask. 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: Most of the fields can be edited by project owners or administrators, in the Weblate interface.

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*
- **Varios complementos**
 - *Detección de componentes*
 - *Generador de estadísticas*
 - *Executing scripts from add-on*

Las variables siguientes están disponibles en las plantillas de componente:

```
{{ language_code }} Código de idioma
{{ language_name }} Nombre del idioma
{{ component_name }} Nombre de componente
{{ component_slug }} «Slug» del componente
{{ project_name }} Nombre del proyecto
{{ project_slug }} «Slug» del proyecto
{{ url }} URL de traducción
{{ filename }} Nombre de archivo de traducción
{{ stats }} Translation stats, this has further attributes, examples below.
{{ stats.all }} Total strings count
{{ stats.fuzzy }} Count of strings needing review
{{ stats.fuzzy_percent }} Percent of strings needing review
{{ stats.translated }} Translated strings count
{{ stats.translated_percent }} Translated strings percent
{{ stats.allchecks }} Number of strings with failing checks
{{ stats.allchecks_percent }} Percent of strings with failing checks
{{ author }} Author of current commit, available only in the commit scope.
{{ addon_name }} Name of currently executed add-on, available only in the add-on commit message.
```

The following variables are available in the repository browser or editor templates:

```
{{branch}} current branch
{{line}} line in file
{{filename}} filename, you can also strip leading parts using the parentdir filter, for example {{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|striptext }}
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* add-on.

To use the add-on, you first need to create a component for one translation file (choose the one that is the least likely to be renamed or removed in future), and install the add-on on this component.

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

Ver también:

Órdenes de gestión, Detección de componentes

2.8 Definiciones de idioma

Para presentar adecuadamente cada una de las traducciones, es necesario contar con información sobre el nombre del idioma, la dirección del texto, la definición de los plurales y el código de lengua.

2.8.1 Procesamiento de códigos de idioma

While parsing translations, Weblate attempts to map language code (usually the ISO 639-1 one) from the *Máscara de archivos* to any existing language object.

You can further adjust this mapping at project level by *Alias de idiomas*.

If no exact match can be found, an attempt will be made to best fit it into an existing language. Following steps are tried:

- Búsquedas sin distinción de mayúsculas y minúsculas.
- Normalización de guiones bajos y guiones.
- 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 *Cambiar las definiciones de idioma*) and report it to the issue tracker (see *Contribuir con Weblate*), so that the proper definition can be added to the upcoming Weblate release.

Consejo: In case you see something unwanted as a language, you might want to adjust *Filtro de idioma* to ignore such file when parsing translations.

Ver también:

Código de idioma, Añadir traducciones nuevas

2.8.2 Cambiar las definiciones de idioma

Es posible cambiar las definiciones de idioma en la interfaz de idiomas (URL `/languages/`).

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

2.8.3 Definiciones de idioma incorporadas

Definitions for about 600 languages are included in Weblate and the list is extended in every release. Whenever Weblate is upgraded (more specifically whenever **weblate 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:

Extending built-in language definitions

2.8.4 Ambiguous language codes and macrolanguages

In many cases it is not a good idea to use macrolanguage code for a translation. The typical problematic case might be Kurdish language, which might be written in Arabic or Latin script, depending on actual variant. To get correct behavior in Weblate, it is recommended to use individual language codes only and avoid macrolanguages.

Ver también:

Macrolanguages definition, List of macrolanguages

2.8.5 Definiciones de idioma

Each language consists of following fields:

Código de idioma

Code identifying the language. Weblate prefers two letter codes as defined by [ISO 639-1](#), but uses [ISO 639-2](#) or [ISO 639-3](#) codes for languages that do not have two letter code. It can also support extended codes as defined by [BCP 47](#).

Ver también:

Procesamiento de códigos de idioma, 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*). The document-based formats (for example *Formato OpenDocument*) start with a copy of the source document and all strings marked as needing editing. In some situations this really doesn't depend on the format, but rather on the framework you use to handle the translation (for example with *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.

Linux style Locales as used by Linux, uses legacy codes for Chinese and POSIX style notation.

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, Procesamiento de códigos de idioma

2.9 Regionalización continua

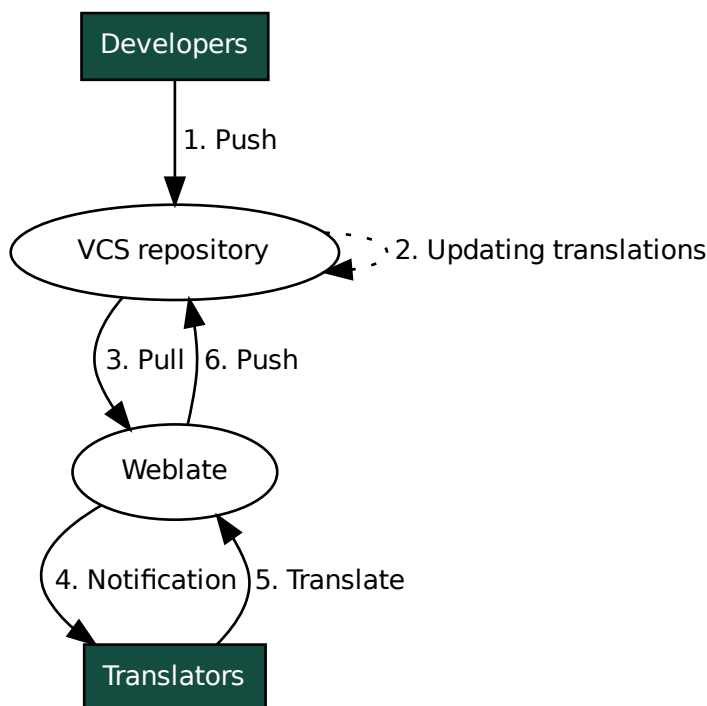
There is infrastructure in place so that your translation closely follows development. This way translators can work on translations the entire time, instead of working through huge amount of new text just prior to release.

Ver también:

`/devel/integration` describes basic ways to integrate your development with Weblate.

This is the process:

1. Developers make changes and push them to the VCS repository.
2. Optionally the translation files are updated (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 repository settings for 'WeblateOrg / hello'. The 'Settings' tab is selected, and the 'Webhooks' section is active. The 'Add webhook' form is displayed with the following configuration:

- Payload URL:** `https://hosted.weblate.org/hooks/github/`
- Content type:** `application/x-www-form-urlencoded`
- Secret:** (empty field)
- SSL verification:** Disabled (button: `Disable SSL verification`)
- Which events would you like to trigger this webhook?**
 - ☒ Just the push event.
 - ☐ Send me **everything**.
 - ☐ Let me select individual events.
- Active:** ☒ (We will deliver event details when this hook is triggered.)
- Add webhook** (button)

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/`.

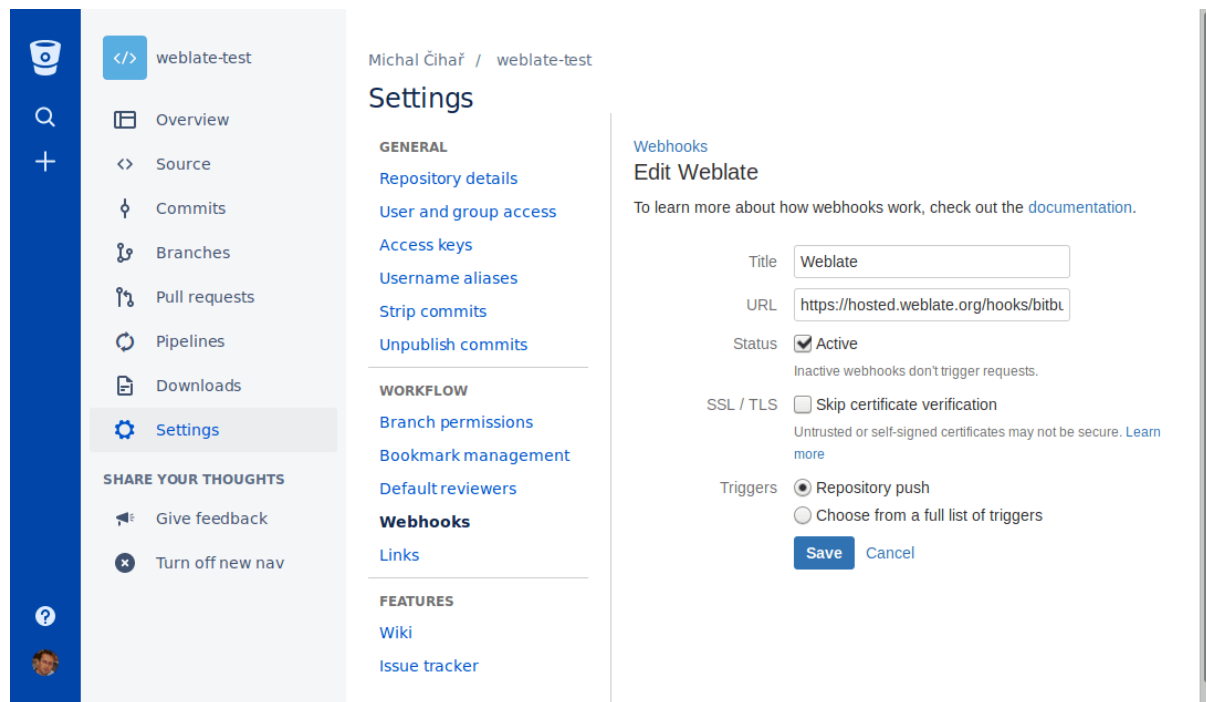
You can leave other values at default settings (Weblate can handle both content types and consumes just the *push* event).

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 the 'fedora PAGURE' logo, a 'Browse' button, a 'Create' dropdown, and a user profile icon. Below this, a secondary bar shows 'New Issue', 'Open PR', 'Fork', and 'Clone' buttons. The main navigation tabs are 'Source', 'Issues' (0), 'Pull Requests' (0), 'Stats', and 'Settings' (active). The left sidebar lists various settings categories: Project Settings, Project Details, Default Branch, Private Web Hook Key, API Keys, Project Options (selected), Public Notifications, Users & Groups, Deploy Keys, Hooks, Priorities, Roadmap, Close Status, Custom Issue Fields, Reports, Tags, Quick Replies, Regenerate Repos, Give Project, and Delete Project. The 'Project Options' section contains a list of checkboxes for various features: 'Activate always merge', 'Activate disable non fast-forward merges', 'Activate Enforce signed-off commits in pull-request', 'Activate fedmsg notifications' (checked), 'Activate Issue tracker' (checked), 'Activate Issue tracker read only', 'Activate Issues default to private', 'Activate Minimum score to merge pull-request' (set to -1), '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' (checked), and 'Activate stomp notifications' (checked). Below these options is a text input for 'Activate Web-hooks' with the value 'https://hosted.weblate.org/hooks/pagure/'. There are 'Update' and 'Test web-hook' buttons. At the bottom, a 'Learn more about' section lists links for '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 pull requests*, *GitLab merge requests*, *Pagure merge requests* 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 pull requests</i>	<i>empty</i>	<i>empty</i>
GitHub pull request from branch	<i>GitHub pull requests</i>	SSH URL ¹	Nombre de la rama
GitLab merge request from fork	<i>GitLab merge requests</i>	<i>empty</i>	<i>empty</i>
GitLab merge request from branch	<i>GitLab merge requests</i>	SSH URL ^{Página 277, 1}	Nombre de la rama
Pagure merge request from fork	<i>Pagure merge requests</i>	<i>empty</i>	<i>empty</i>
Pagure merge request from branch	<i>Pagure merge requests</i>	SSH URL ^{Página 277, 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 Interactuar con otros

Weblate facilita la interacción con otras herramientas mediante su API.

Ver también:

[API REST de Weblate](#)

2.9.4 Consignas diferidas

The behaviour of Weblate is to group commits from the same author into one commit if possible. This greatly reduces the number of commits, however you might need to explicitly tell it to do the commits in case you want to get the VCS repository in sync, e.g. for merge (this is by default allowed for the *Managers* group, see *[Lista de privilegios y roles incorporados](#)*).

Los cambios en esta modalidad se consignan una vez que cualquiera de estas condiciones se cumpla:

- Alguien más modifica una cadena ya modificada.
- Se produce una fusión desde el origen ascendente.
- Se solicita explícitamente una consigna.
- Change is older than period defined as *[Antigüedad de cambios por consignar](#)* on *[Configuración de componentes](#)*.

Consejo: Commits are created for every component. So in case you have many components you will still see lot of commits. You might utilize *[Concentrar consignas de Git](#)* add-on in that case.

If you want to commit changes more frequently and without checking of age, you can schedule a regular task to perform a commit:

```
CELERY_BEAT_SCHEDULE = {
    # Unconditionally commit all changes every 2 minutes
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Omitting 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,
    }
}
```

2.9.5 Procesar el repositorio con secuencias

The way to customize how Weblate interacts with the repository is *Complementos*. Consult *Executing scripts from add-on* for info on how to execute external scripts through add-ons.

2.9.6 Mantener iguales las traducciones entre los componentes

Once you have multiple translation components, you might want to ensure that the same strings have same translation. This can be achieved at several levels.

Propagación de traducciones

With *Permitir propagación de traducciones* enabled (what is the default, see *Configuración de componentes*), all new translations are automatically done in all components with matching strings. Such translations are properly credited to currently translating user in all components.

Nota: The translation propagation requires the key to be match for monolingual translation formats, so keep that in mind when creating translation keys.

Comprobación de coherencia

The *Incoherente* check fires whenever the strings are different. You can utilize this to review such differences manually and choose the right translation.

Traducción automática

Automatic translation based on different components can be way to synchronize the translations across components. You can either trigger it manually (see *Traducción automática*) or make it run automatically on repository update using add-on (see *Traducción automática*).

2.10 Licensing translations

You can specify which license translations are contributed under. This is especially important to do if translations are open to the public, to stipulate what they can be used for.

You should specify *Configuración de componentes* license info. You should avoid requiring a contributor license agreement, though it is possible.

2.10.1 License info

Upon specifying license info (license name and URL), this info is shown in the translation info section of the respective *Configuración de componentes*.

Usually this is best place to post licensing info if no explicit consent is required. If your project or translation is not libre you most probably need prior consent.

2.10.2 Acuerdo de contribuidor

If you specify a contributor license agreement, only users who have agreed to it will be able to contribute. This is a clearly visible step when accessing the translation:

The 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%' badge. 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 'Languages' tab is active, showing a table of translations:

Language	Translated	Unfinished	Unfinished words	Checks	Suggestions	Comments
Czech 🇨🇪 GPL-3.0	✓					
Hebrew 🇮🇱 GPL-3.0	✓					
Hungarian 🇮🇪 GPL-3.0	81%	4	5			
English 🇬🇧 GPL-3.0	✓					

At the bottom of the languages section is a 'Start new translation' button. The footer includes 'Powered by Weblate 4.12' and links for 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

The entered text is formatted into paragraphs and external links can be included. HTML markup can not be used.

2.10.3 User licenses

Any user can review all translation licenses of all public projects on the instance from their profile:

Licenses

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:

- [WeblateOrg/Language names](#)

Licenses for individual translations

GNU General Public License v3.0 or later GPL-3.0 ⓘ

[WeblateOrg/WebateOrg](#) [WeblateOrg/Djangojs](#) [WeblateOrg/Django](#) [WeblateOrg/Language names](#)

MIT License MIT ⓘ

[WeblateOrg/Android](#)

Powered by Weblate 4.12 [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* 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).

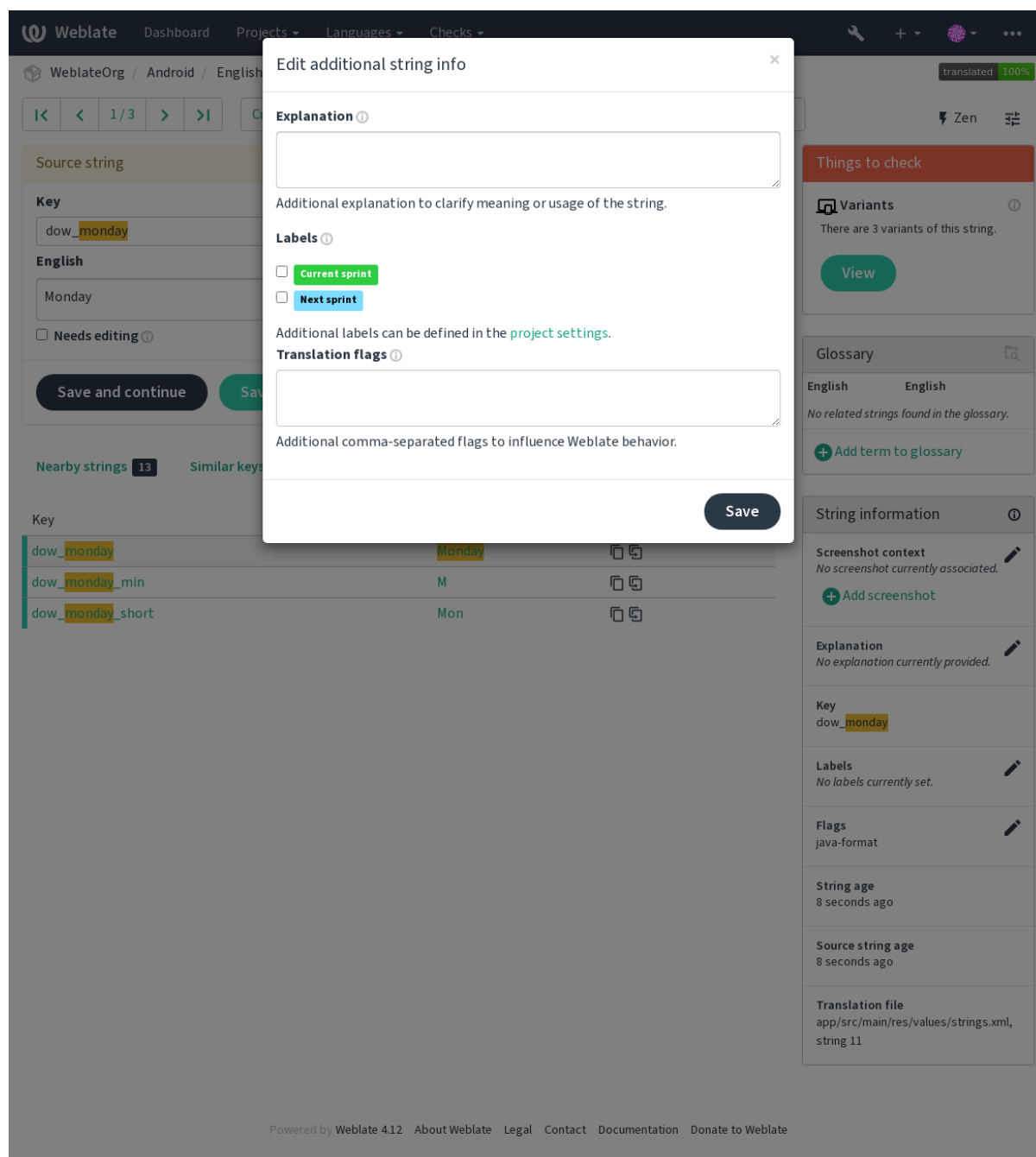
Nota: Once automatic acceptance is set up, normal users lose the privilege to directly save translations or accept suggestions. This can be overridden with the *Edit string when suggestions are enforced permission*.

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 and continue Save and stay Suggest Skip

Nearby strings 26 Comments Other languages 4 History

Context	English	Czech	Actions
	Files	Soubory	
	Automatic translation	Automatický překlad	
	Add new translation string	Add new translation string	
	Translation status	Stav překladu	
	%{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 překlady v projektu na tuto součást. Může být užitečný pro sloučení překladů z jiné větve, opravu nekonzistentních překladů nebo překlad nové součásti pomocí překladové paměti.	
	Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.	Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.	
	You can add new translation string here, it will automatically appear in all translations.	Zde můžete přidat nový řetězec k překladu, automaticky se objeví ve všech jazycích.	
	The uploaded file will be merged with the current translation. In case you want to overwrite already translated strings, don't forget to enable it.	Nahráný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.	
	The uploaded file will be merged with the current translation.	Nahráný soubor bude sloučen se stávajícími překlady.	
	The fulltext search might not work properly as the fulltext index for this translation is not yet up to date.	Fulltextové vyhledávání nemusí fungovat správně, protože fulltextový index pro tento překlad ještě není plně zpracován.	
	Review	Kontrola	
	Review translations touched by other users.	Zkontrolovat překlady od ostatních uživatelů.	
	Start review	Začít kontrolu	
	Percent	Procenta	
	Total	Celkem	
	Failing check	Neúspěšných kontrol	
	Last activity	Poslední aktivity	
	Last change	Poslední změna	
	Last author	Poslední autor	
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 strojový weblate.org

translation strojový překlad

project projekt weblate.org

+ 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

a second ago

Source string age

2 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.

Customization of quality checks and other Weblate behavior, see *Personalizar el comportamiento mediante indicadores*.

The string flags are also inherited from the *Indicadores de traducción* at *Configuración de componentes* and flags from the translation file (see *Formatos de archivo admitidos*).

Ver también:

Comprobaciones de calidad, *Personalizar el comportamiento mediante indicadores*

Explicación

Distinto en la versión 4.1: 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:

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

< > 11 / 26 All strings Position and priority Zen

Translation

Explanation

Help text for automatic translation tool

English

Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

☐ Needs editing

Save and continue Save and stay Suggest Skip

Nearby strings 26 Comments Automatic suggestions Other languages 4 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.	Weblate Translation Memory (Project: weblateorg/django) Weblate Translation Memory (Shared: weblateorg/django) Weblate (WeblateOrg/Django)	100 %	Copy Copy and save

Glossary

English	Czech	
machine translation	strojový překlad	WeblateOrg
project	projekt	WeblateOrg

+ Add term to glossary

String information

Screenshot context

```

Search string
...
<div>
  ...
</div>

```

+ 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

5 seconds ago

Source string age

5 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/djan-go.po, string 11

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:

WebateDashboardProjectsLanguagesChecks

WebateOrg / Django / Screenshots / Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			
Screenshot is shown to add visual context for all listed source strings.			

Assign source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			

Source string search

Search

Automatically recognize

Image

Source string

Hello, world!↵

OneOrangutan has %d banana.↵

OtherOrangutan has %d bananas.↵

Try Weblate at <http://demo.weblate.org/>!↵

Thank you for using Weblate.

Screenshot is shown to add visual context for all listed source strings.

Edit screenshot

Screenshot name

Automatic translation

Image

Currently: screenshots/screenshot.png

Change:

Choose File

No file chosen

Upload JPEG or PNG images up to 2000x2000 pixels.

Save

Screenshot details

Created	now
Uploaded by	testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

2.12 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 © 2012-2022 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 *Custom quality checks, add-ons and auto-fixes*.

2.12.2 Personalizar el comportamiento mediante indicadores

You can fine-tune the Weblate behavior by using flags. This can be done on the source string level (see *Información adicional sobre las cadenas de origen*), or in the *Configuración de componentes (Indicadores de traducción)*. Some file formats also allow to specify flags directly in the format (see *Formatos de archivo admitidos*).

The flags are comma-separated, 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*.

dos-eol Uses DOS end-of-line markers instead of Unix ones (`\r\n` instead of `\n`).

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*.

icu-flags:FLAGS Define flags for customizing the behavior of the *MessageFormat de ICU* quality check.

icu-tag-prefix:PREFIX Set a required prefix for XML tags for the *MessageFormat de ICU* quality check.

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 Marcar esta cadena como una variante de cadena con origen coincidente. Ver variants.

regex:REGEX Expresión regular utilizada para relacionar traducciones; vea *Expresión regular*.

forbidden Indica una traducción prohibida en un glosario, ver *Traducciones prohibidas*.

strict-same Hace que «Traducción no modificada» evite la lista de palabras incorporada; vea *Traducción no modificada*.

check-glossary Activar el control de calidad *No se ajusta al glosario*.

angularjs-format Activar el control de calidad *Cadena de interpolación de AngularJS*.

c-format Activar el control de calidad *Formato C*.

c-sharp-format Activar el control de calidad *Formato C#*.

es-format Activar el control de calidad *Literales de plantilla ECMAScript*.

i18next-interpolation Activar el control de calidad *Interpolación con i18next*.

icu-message-format Enable the *MessageFormat de ICU* quality check.

java-format Activar el control de calidad *Formato Java*.

java-messageformat Activar el control de calidad *MessageFormat de Java*.

javascript-format Activar el control de calidad *Formato JavaScript*.

lua-format Activar el control de calidad *Formato Lua*.

object-pascal-format Activar el control de calidad *Formato de Object Pascal*.

percent-placeholders Activar el control de calidad *Sustitutorios con signo de porcentaje*.

perl-format Activar el control de calidad *Formato Perl*.

php-format Activar el control de calidad *Formato PHP*.

python-brace-format Activar el control de calidad *Formato de llaves de Python*.

python-format Activar el control de calidad *Formato Python*.

qt-format Activa el control de calidad *Formato Qt*.

qt-plural-format Activa el control de calidad *Formato de plurales de Qt*.

ruby-format Activa el control de calidad *Formato Ruby*.

scheme-format Activa el control de calidad *Formato Scheme*.

vue-format Activa el control de calidad *Formato de Vue I18n*.

md-text Trata el texto como un documento Markdown. Activa el control de calidad *Enlaces de Markdown*, *Referencias de Markdown*, y *Sintaxis de Markdown*.

safe-html Activa el control de calidad *HTML inseguro*.

url La cadena debe consistir de solo una URL. Activa el control de calidad *URL*.

ignore-all-checks Ignore all quality checks.

ignore-bbcode Saltar el control de calidad *Marcación BBCode*.

ignore-duplicate Omite el control de calidad *Palabras consecutivas duplicadas*.

ignore-check-glossary Omite el control de calidad *No se ajusta al glosario*.

ignore-double-space Omite el control de calidad *Espacio duplicado*.

ignore-angularjs-format Omite el control de calidad *Cadena de interpolación de AngularJS*.

ignore-c-format Omite el control de calidad *Formato C*.

ignore-c-sharp-format Omite el control de calidad *Formato C#*.

ignore-es-format Omite el control de calidad *Literales de plantilla ECMAScript*.

ignore-i18next-interpolation Omite el control de calidad *Interpolación con i18next*.

ignore-icu-message-format Skip the *MessageFormat de ICU* quality check.

ignore-java-format Omite el control de calidad *Formato Java*.

ignore-java-messageformat Omite el control de calidad *MessageFormat de Java*.

ignore-javascript-format Omite el control de calidad *Formato JavaScript*.

ignore-lua-format Skip the *Formato Lua* quality check.

ignore-object-pascal-format Skip the *Formato de Object Pascal* quality check.

ignore-percent-placeholders Skip the *Sustitutorios con signo de porcentaje* quality check.

ignore-perl-format Skip the *Formato Perl* quality check.

ignore-php-format Skip the *Formato PHP* quality check.

ignore-python-brace-format Skip the *Formato de llaves de Python* quality check.

ignore-python-format Skip the *Formato Python* quality check.

ignore-qt-format Skip the *Formato Qt* quality check.

ignore-qt-plural-format Skip the *Formato de plurales de Qt* quality check.

ignore-ruby-format Skip the *Formato Ruby* quality check.

ignore-scheme-format Skip the *Formato Scheme* quality check.

ignore-vue-format Skip the *Formato de Vue I18n* quality check.

ignore-translated Skip the *Se había traducido* quality check.

ignore-inconsistent Skip the *Incoherente* quality check.

ignore-kashida Skip the *Kashida utilizado* quality check.

ignore-md-link Skip the *Enlaces de Markdown* quality check.

ignore-md-reflink Skip the *Referencias de Markdown* quality check.

ignore-md-syntax Skip the *Sintaxis de Markdown* quality check.

ignore-max-length Skip the *Longitud máxima de la traducción* quality check.

ignore-max-size Skip the *Tamaño máximo de la traducción* quality check.

ignore-escaped-newline Skip the *\n desiguales* quality check.

ignore-end-colon Skip the *Dos puntos desiguales* quality check.

ignore-end-ellipsis Skip the *Puntos suspensivos desiguales* quality check.

ignore-end-exclamation Skip the *Signo de exclamación desigual* quality check.

ignore-end-stop Skip the *Punto final desigual* quality check.

ignore-end-question Skip the *Signo de interrogación desigual* quality check.

ignore-end-semicolon Skip the *Punto y coma desigual* quality check.

ignore-newline-count Skip the *Salto de renglón desiguales* quality check.

ignore-plurals Skip the *Faltan plurales* quality check.

ignore-placeholders Skip the *Sustitutorios* quality check.

ignore-punctuation-spacing Skip the *Espaciado de puntuación* quality check.

ignore-regex Skip the *Expresión regular* quality check.

ignore-same-plurals Skip the *Mismos plurales* quality check.

ignore-begin-newline Skip the *Salto de renglón al inicio* quality check.

ignore-begin-space Skip the *Espacios iniciales* quality check.

ignore-end-newline Skip the *Salto de renglón al final* quality check.

ignore-end-space Skip the *Espacio al final* quality check.

ignore-same Skip the *Traducción no modificada* quality check.

ignore-safe-html Omite la comprobación de calidad *HTML inseguro*.

ignore-url Omite la comprobación de calidad *URL*.

ignore-xml-tags Omite la comprobación de calidad *Marcación XML*.

ignore-xml-invalid Omite la comprobación de calidad *Sintaxis XML*.

ignore-zero-width-space Omite la comprobación de calidad *Espacio de anchura cero*.

ignore-ellipsis Omite la comprobación de calidad *Puntos suspensivos*.

ignore-icu-message-format-syntax Skip the *Sintaxis MessageFormat de ICU* quality check.

ignore-long-untranslated Omite la comprobación de calidad *Largamente no traducida*.

ignore-multiple-failures Omite la comprobación de calidad *Varias comprobaciones fallidas*.

ignore-unnamed-format Omite la comprobación de calidad:ref:check-unnamed-format.

ignore-optional-plural Omite la comprobación de calidad *No pluralizada*.

Nota: Generalmente la regla se denomina `ignore-*` para cualquier comprobación, usa su identificador, así puedes utilizarlo incluso para tus comprobaciones personalizadas.

Estas flags se entienden tanto en la configuración de *Configuración de componentes*, por la configuración de la cadena de origen y en el propio archivo de traducción (por ejemplo en GNU gettext).

2.12.3 Forzar comprobaciones

Nuevo en la versión 3.11.

Puedes configurar una lista de comprobaciones que no se pueden ignorar estableciendo *Comprobaciones obligatorias* en *Configuración de componentes*. No se pueden ignorar cada una de las comprobaciones enumeradas en la interfaz de usuario y cualquier cadena que falle esta comprobación se marca como: *Needs editing* (ver *Estados de traducción*).

2.12.4 Gestionar tipos de letra


Nuevo en la versión 3.7.

Consejo: Las fuentes cargadas en Weblate se usan únicamente a efectos de comprobación *Tamaño máximo de la traducción*, no tienen efecto en la interfaz de usuario de Weblate.

La comprobación *Tamaño máximo de la traducción* utilizada para calcular las dimensiones del texto renderizado necesita que la fuente se cargue en Weblate y se seleccione mediante una flag de traducción (ver *Personalizar el comportamiento mediante indicadores*).

La herramienta de gestión de fuentes de Weblate en *Fonts* bajo el menú *Manage* de tu proyecto de traducción proporciona una interfaz para cargar y gestionar las fuentes. Se pueden cargar fuentes TrueType u OpenType, configurar grupos de fuentes y utilizarlas en la comprobación.

Los grupos de fuentes permiten definir diferentes fuentes para distintos idiomas, lo que suele ser necesario para los idiomas no latinos:





 Weblate

[Dashboard](#)

[Projects](#)

[Languages](#)

[Checks](#)

[WeblateOrg](#) / [Font groups](#) / [default-font](#)

Font group

Name	default-font		
Default font	Source Sans 3 Bold		
Japanese	language override	Droid Sans Fallback Regular	Remove
Korean	language override	Droid Sans Fallback Regular	Remove
Delete			

Add language override

Language

Font

[Save](#)

Edit font group

Font group name

default-font

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

Source Sans 3 Bold

Default font is used unless per language override matches.

[Save](#)

Powered by Weblate 4.12

[About Weblate](#)


[Legal](#)

[Contact](#)


[Documentation](#)

[Donate to Weblate](#)

Los grupos de fuentes se identifican por su nombre, que no puede contener espacios en blanco ni caracteres especiales, para que pueda utilizarse fácilmente en la definición de la comprobación:

 Weblate

DashboardProjectsLanguagesChecks

 WeblateOrg / Fonts

Font groupsFonts

Group name	Default font	Language overrides	
default-font	Source Sans 3 Bold	Japanese: Droid Sans Fallback Regular Korean: Droid Sans Fallback Regular	Edit

Add font group

Font group name

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.


Default font

Default font is used unless per language override matches.


Save


Powered by Weblate 4.12About WeblateLegalContactDocumentationDonate to Weblate

El tipo de letra y el estilo se reconocen automáticamente después de cargarlos:

 Weblate

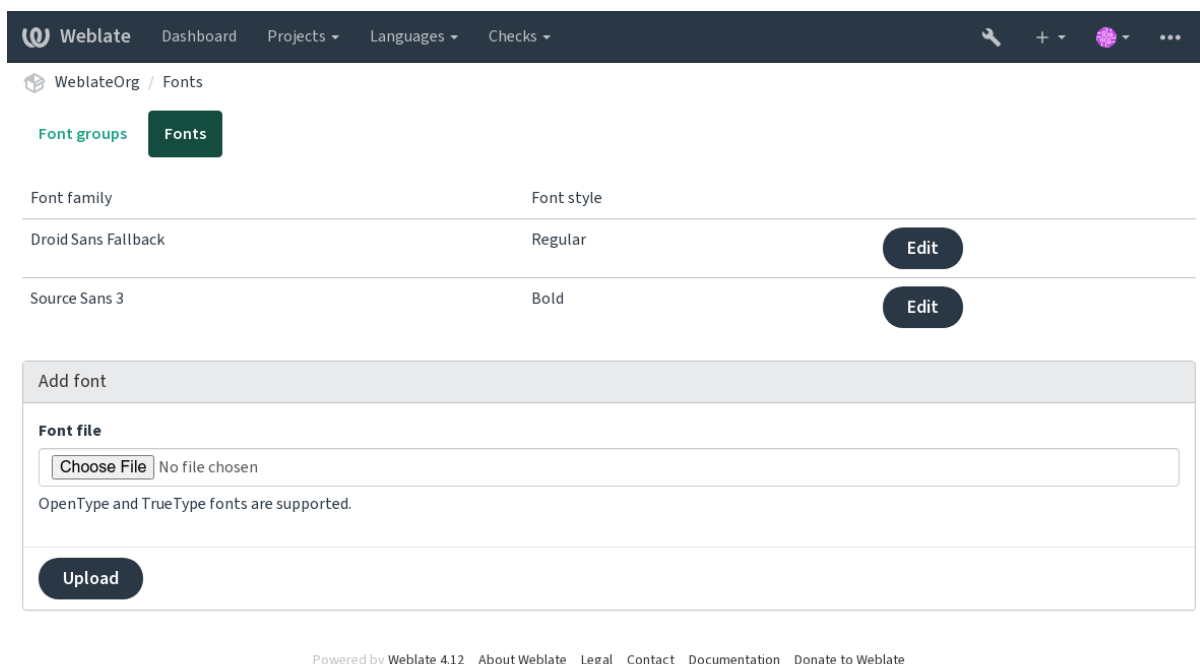
DashboardProjectsLanguagesChecks

 WeblateOrg / Fonts / Droid Sans Fallback Regular

Font	
Font family	Droid Sans Fallback
Font style	Regular
File size	3939852
Created	now
Uploaded by	 testuser
Used in groups	
Delete	

Powered by Weblate 4.12About WeblateLegalContactDocumentationDonate to Weblate

Puedes tener varias fuentes cargadas en Weblate:



Para utilizar las fuentes que comprueban la longitud de la cadena, pásale los indicadores adecuados (ver [Personalizar el comportamiento mediante indicadores](#)). Posiblemente necesitarás los siguientes:

max-size:500 Defines maximal width in pixels.

font-family:ubuntu Define el grupo de fuentes a utilizar especificando su identificador.

font-size:22 Defines font size in pixels.

2.12.5 Escribir los propios controles

Se han incorporado una amplia gama de controles de calidad, (ver [Comprobaciones de calidad](#)), aunque es posible que no cubran todo lo que quieres comprobar. Se puede ajustar la lista de comprobaciones realizadas mediante `CHECK_LIST`, y también puedes añadir controles personalizados.

1. Subclase del `weblate.checks.Check`
2. Defina algunos atributos.
3. Implementa el `check` (si quieres tratar los plurales en tu código) o el método `check_single` (que lo hace por ti).

He aquí algunos ejemplos:

Para instalar controles personalizados, proporciona una ruta muy clara de clase Python en `CHECK_LIST`, ver [Custom quality checks, add-ons and auto-fixes](#).

Comprueba que el texto de la traducción no contiene «foo»

Esta es una comprobación bastante simple que se limita a revisar si a la traducción le falta la cadena «tal».

```
#
# Copyright © 2012–2022 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
```

(continué en la próxima página)

(proviene de la página anterior)

```
# 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/>.
#
"""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
```

Comprueba que los plurales del texto traducido al checo son diferentes

Comprobación que utiliza la información de idioma para cerciorarse de que las dos formas de plural en checo no sean iguales.

```
#
# Copyright © 2012-2022 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."""
```

(continué en la próxima página)

(proviene de la página anterior)

```
from django.utils.translation import gettext_lazy as _

from weblate.checks.base import TargetCheck

class PluralCzechCheck(TargetCheck):

    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_target_unit(self, sources, targets, unit):
        if self.is_language(unit, ("cs",)):
            return targets[1] == targets[2]
        return False

    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). In case you are on a free instead of a paid plan, you have to use `https://api-free.deepl.com/` instead of `https://api.deepl.com/` You can adjust both parameters by `MT_DEEPL_API_URL`.

Ver también:

`MT_DEEPL_KEY`, `MT_DEEPL_API_URL`, [DeepL website](#), [DeepL pricing](#), [DeepL API documentation](#)

2.13.6 LibreTranslate

Nuevo en la versión 4.7.1.

LibreTranslate is a free and open-source service for machine translations. The public instance requires an API key, but LibreTranslate can be self-hosted and there are several mirrors available to use the API for free.

Turn on this service by adding `weblate.machinery.libretranslate.LibreTranslateTranslation` to `MT_SERVICES` and set `MT_LIBRETRANSLATE_API_URL`. If your instance requires an API key, you must also set `MT_LIBRETRANSLATE_KEY`.

Ver también:

`MT_LIBRETRANSLATE_KEY`, `MT_LIBRETRANSLATE_API_URL`, LibreTranslate website, LibreTranslate repository, LibreTranslate mirrors

2.13.7 Glosbe

Free dictionary and translation memory for almost every living language.

The API is gratis to use, but usage of the translations is subject to the license of the used data source. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Turn on this service by adding `weblate.machinery.glosbe.GlosbeTranslation` to `MT_SERVICES`.

Ver también:

Glosbe website

2.13.8 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

2.13.9 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

2.13.10 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

2.13.11 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

2.13.12 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`

2.13.13 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](#)

2.13.14 Traducción automática por API de NetEase Sight

Nuevo en la versión 3.3.

Machine translation service provided by 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](#)

2.13.15 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](#)

2.13.16 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](#)

2.13.17 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](#)

2.13.18 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`.

2.13.19 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.

2.13.20 SAP Translation Hub

Machine translation service provided by SAP.

You need to have a SAP account (and the SAP Translation Hub enabled in the SAP Cloud Platform) to use this service.

Turn on this service by adding `weblate.machinery.saptranslationhub.SAPTranslationHub` to `MT_SERVICES` and set the appropriate access to either the sandbox or the production 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

2.13.21 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 © 2012-2022 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/>.
#
"""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 your own class in `MT_SERVICES` and Weblate will start using that.

2.14 Complementos

Nuevo en la versión 2.19.

Add-ons provide ways to customize and automate the translation workflow. Admins can add and manage add-ons from the *Manage* ↓ *Add-ons* menu of each respective translation component.

Consejo: You can also configure add-ons using [API](#), [DEFAULT_ADDONS](#), or [install_addon](#).



Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / Language names / Add-ons

Installed add-ons


There are no add-ons currently installed.

Available add-ons


Automatic translation

Automatically translates strings using machine translation or other components.


Install


Add missing languages

Ensures a consistent set of languages is used for all components within a project.

project wide


Install


Component discovery

Automatically adds or removes project components based on file changes in the version control system.


repository wide

Install


Bulk edit


Bulkedit flags, labels, or states of strings.

Install


Statistics generator


Generates a file containing detailed info about the translation status.

Install


Prefill translation with source


Fills in translation strings with source string.

Install


Pseudolocale generation


Generates a translation by adding prefix and suffix to source strings automatically.

Install


Contributors in comment


Updates the comment part of the PO file header to include contributor names and years of contributions.

Install


Customize gettext output


Allows customization of gettext output behavior, for example line wrapping.

Install


Generate MO files


Automatically generates a MO file for every changed PO file.

Install


Update PO files to match POT (msgmerge)

Updates all PO files (as configured by "File mask") to match the POT file (as configured by "Template for new translations") using msgmerge.


Install


Squash Git commits

Squash Git commits prior to pushing changes.

repository wide


Install


Stale comment removal

Set a timeframe for removal of comments.

project wide

Install


Stale suggestion removal

Set a timeframe for removal of suggestions.

project wide

Install

Some add-ons will ask for additional configuration during installation.

2.14.1 Built-in add-ons

Traducción automática

Nuevo en la versión 3.9.

Add-on ID `weblate.autotranslate.autotranslate`

Configuración

mo- de	Modo de tra- ducción automa- tizada	Available choices: suggest – Add as suggestion translate – Add as translation fuzzy – Add as needing edit
fil- ter	Filtro de búsqueda	Observe que traducir todas las cadenas descartará todas las traducciones existentes. Available choices: all – All strings nottranslated – Untranslated strings todo – Unfinished strings fuzzy – Strings marked for edit check:inconsistent – Failing check: Inconsistent
au- to_s	Fuente de la tra- ducción automá- tica	Available choices: others – Other translation components mt – Machine translation
com- po- nent	Compo- nentes	Introduzca el componente que deberá utilizarse como fuente. Manténgalo vacío para utilizar todos los componentes del proyecto actual.
en- gi- nes	Motores de tra- ducción automá- tica	
th- res- hold	Umbral de pun- tuación	

Triggers component update, daily

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.

Add-on ID `weblate.cdn.cdnjs`

Configuración

thres- hold	Umbral de traduc- ción	Umbral para la inclusión de traducciones.
css_sele	Selector de CSS	Selector de CSS para detectar elementos regionalizables.
co- okie_name	Nombre de la «co- okie» de idioma	Nombre de la «cookie» que almacena la preferencia de idioma.
files	Extraer cadenas desde archivos HTML	Lista de nombres de archivo en el repositorio actual o URL remotos que analizar en busca de cadenas traducibles.

Triggers `daily`, `repository post-commit`, `repository post-update`

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.

Add-on ID `weblate.cleanup.blank`

Configuración *This add-on has no configuration.*

Triggers `repository post-commit`, `repository post-update`

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

Add-on ID `weblate.cleanup.generic`

Configuración *This add-on has no configuration.*

Triggers `repository pre-commit`, `repository post-update`

Actualiza todos los archivos de traducción para que coincidan con el archivo de base monolingüe. Para la mayoría de los formatos de archivo, esto implica quitar las claves de traducción obsoletas que ya no están en el archivo de base.

Ver también:

Does Weblate update translation files besides translations?

Adición de idiomas faltantes

Add-on ID `weblate.consistency.languages`

Configuración *This add-on has no configuration.*

Triggers daily, repository post-add

Garantiza que en todos los componentes de un proyecto se utilice el mismo conjunto de idiomas.

Missing languages are checked once every 24 hours, and when new languages are added in Weblate.

Unlike most others, this add-on affects the whole project.

Consejo: Auto-translate the newly added strings with *Traducción automática*.

Detección de componentes

Add-on ID `weblate.discovery.discovery`

Configuración

match	Expresión regular utilizada para relacionar con los archivos de traducción	
file_format	Formato de archivo	
name_template	Personalizar el nombre del componente	
base_file_template	Definir el nombre del archivo de base por idioma	Dejar vacío para archivos de traducción bilingües.
new_base_file_template	Definir el archivo de base para traducciones nuevas	Nombre del archivo utilizado para traducciones nuevas. Para gettext seleccione un archivo .pot.
language_regex	Filtro de idioma	Regular expression to filter translation files against when scanning for file mask.
copy_addons	Clonar complementos del componente principal a los nuevos	
remove	Remove components for inexistent files	
confirm	Confirmo que las coincidencias anteriores parecen correctas	

Triggers repository post-update

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 add-on 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

+

WeblateOrg

Language names

Add-ons

Component discovery

Configure add-on

Please review and confirm the matched components.

Component	Matched files
The following components would be created	
Djangojs	<div>weblate/locale/cs/LC_MESSAGES/djangojs.po (cs)</div> <div>weblate/locale/hu/LC_MESSAGES/djangojs.po (hu)</div> <div>weblate/locale/he/LC_MESSAGES/djangojs.po (he)</div>
Django	<div>weblate/locale/cs/LC_MESSAGES/django.po (cs)</div> <div>weblate/locale/hu/LC_MESSAGES/django.po (hu)</div> <div>weblate/locale/he/LC_MESSAGES/django.po (he)</div>

☐ 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 file mask.

☒ Clone add-ons from the main component to the newly created ones

☐ Remove components for inexistent files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description
(?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 4.12 About Weblate Legal Contact Documentation Donate to Weblate

308

Capítulo 2. Documentación para administradores

Consejo: Component discovery add-on 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.

Add-on ID `weblate.flags.bulk`

Configuración

q	Consulta	
state	Estado que definir	Available choices: -1 – Do not change 10 – Needs editing 20 – Translated 30 – Approved
add_flags	Indicadores de traducción que añadir	
remove_flags	Indicadores de traducción que quitar	
add_labels	Etiquetas que añadir	
remove_labels	Etiquetas que quitar	

Triggers component update

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 5: Etiquetar cadenas nuevas automáticamente

Consulta de búsqueda	<code>NOT has:label</code>
Etiquetas que añadir	<code>recent</code>

Tabla 6: Marking all Archivos de metadatos de tiendas de aplicaciones changelog strings 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.

Add-on ID `weblate.flags.same_edit`

Configuración *This add-on has no configuration.*

Triggers `unit post-create`

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:

Estados de traducción

Marcar cadenas de origen nuevas como «Necesitan edición»

Add-on ID `weblate.flags.source_edit`

Configuración *This add-on has no configuration.*

Triggers `unit post-create`

Cuando se importa una cadena de origen nueva desde el VCS, se marca como necesaria la edición en Weblate. De esta manera usted puede filtrar y editar cadenas de origen escritas por los desarrolladores con facilidad.

Ver también:

Estados de traducción

Marcar traducciones nuevas como «Necesitan edición»

Add-on ID `weblate.flags.target_edit`

Configuración *This add-on has no configuration.*

Triggers `unit post-create`

Cuando se importa una cadena traducible nueva desde el VCS, se marca como necesaria la edición en Weblate. De esta manera usted puede filtrar y editar las traducciones creadas por los desarrolladores con facilidad.

Ver también:

Estados de traducción

Generador de estadísticas

Add-on ID `weblate.generate.generate`

Configuración

<code>filename</code>	Nombre del archivo generado	
<code>template</code>	Contenido del archivo generado	

Triggers `repository pre-commit`

Genera un archivo con información detallada sobre el estado de la traducción.

Es posible utilizar una plantilla de Django tanto en el nombre de archivo como en el contenido; vea [Marcación de plantilla](#) para obtener una descripción a detalle de la marcación.

Por ejemplo, para generar un archivo resumen para cada una de las traducciones:

Nombre del archivo generado locale/{{ código_de_idioma }}.json

Contenido

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}"
}
```

Ver también:

[Marcación de plantilla](#)

Prerrellenar traducción con original

Nuevo en la versión 4.11.

Add-on ID weblate.generate.prefill

Configuración *This add-on has no configuration.*

Triggers component update, daily

Completa las cadenas de traducción con las de origen.

All untranslated strings in the component will be filled with the source string, and marked as needing edit. Use this when you can not have empty strings in the translation files.

Generación de pseudorregiones

Nuevo en la versión 4.5.

Add-on ID weblate.generate.pseudolocale

Configuración

source	Cadenas de origen	
target	Traducción de destino	Todas las cadenas de esta traducción serán sobrescritas
prefix	Fixed string prefix	
var_prefix	Variable string prefix	
suffix	Fixed string suffix	
var_suffix	Variable string suffix	
var_multiplier	Multiplicador de partes variable	How many times to repeat the variable part depending on the length of the source string.

Triggers component update, daily

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.

Using the variable parts makes it possible to look for strings which might not fit into the user interface after the localization - it extends the text based on the source string length. The variable parts are repeated by length of the text multiplied by the multiplier. For example `Hello world` with variable suffix `_` and variable multiplier of `1` becomes `Hello world_____` - the suffix is repeated once for each character in the source string.

Las cadenas se generarán utilizando el siguiente patrón:

Fixed string prefix Variable string prefix Source string Variable string suffix Fixed string suffix

Consejo: You can use real languages for testing, but there are dedicated pseudolocales available in Weblate - *en_XA* and *ar_XB*.

Consejo: You can use this add-on to start translation to a new locale of an existing language or similar language. Once you add the translation to the component, follow to the add-on. *Example:* If you have *fr* and want to start *fr_CA* translation, simply set *fr* as the source, *fr_CA* as the target, and leave the prefix and suffix blank.

Uninstall the add-on once you have the new translation filled to prevent Weblate from changing the translations made after the copying.

Contribuidores en comentarios

Add-on ID `weblate.gettext.authors`

Configuración *This add-on has no configuration.*

Triggers `repository pre-commit`

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»

Add-on ID `weblate.gettext.configure`

Configuración *This add-on has no configuration.*

Triggers `repository post-add, daily`

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

Add-on ID `weblate.gettext.customize`

Configuración

width	Ajuste de renglones largos	De manera predeterminada, gettext ajusta los renglones a los 77 caracteres y en los saltos de renglón. Con el parámetro <code>--no-wrap</code> ajusta solo en los saltos. Available choices: 77 – Wrap lines at 77 characters and at newlines 65535 – Only wrap lines at newlines -1 – No line wrapping
-------	----------------------------	---

Triggers `storage post-load`

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

Add-on ID `weblate.gettext.linguas`

Configuración *This add-on has no configuration.*

Triggers `repository post-add, daily`

Actualiza el archivo LINGUAS al añadir una traducción nueva.

Generar archivos MO

Add-on ID `weblate.gettext.mo`

Configuración

path	Ruta del archivo MO generado	Si no se especifica, se utilizará la ubicación del archivo PO.
------	------------------------------	--

Triggers `repository pre-commit`

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)

Add-on ID `weblate.gettext.msgmerge`

Configuración

<code>previous</code>	Conservar los msgid anteriores de las cadenas traducidas	
<code>no_location</code>	Quitar ubicaciones de cadenas traducidas	
<code>fuzzy</code>	Usar coincidencia aproximada	

Triggers repository post-update

Actualiza todos los archivos PO (tal como se configuraron mediante *Máscara de archivos*) a juego con el archivo POT (según esté configurado por *Plantilla para traducciones nuevas*) utilizando **msgmerge**.

Triggered whenever new changes are pulled from the upstream repository. Most msgmerge command-line options can be set up through the add-on configuration.

Ver también:

Does Weblate update translation files besides translations?

Concentrar consignas de Git

Add-on ID `weblate.git.squash`

Configuración

<code>squash</code>	Concentración de consignas	Available choices: all – All commits into one language language – Per language file – Per file author – Per author
<code>append_final_message</code>	Añadir renglones finales al mensaje de consigna concentrada	Los renglones finales se asemejan a cabeceras de correo RFC 822 y se añaden a la sección libre de un mensaje de consigna. Tales renglones pueden contener indicaciones como: «Co-authored-by: ...».
<code>commit_message</code>	Mensaje de consigna	Este mensaje de consigna se utilizará en sustitución de los mensajes combinados de las consignas concentradas.

Triggers repository post-commit

Concentrar las consignas de Git antes de enviar los cambios.

Las consignas Git se pueden concentrar antes de enviar los cambios en uno de los modos siguientes:

- Todas las consignas en una sola
- Por idioma
- Por archivo
- Por autor

Se conservarán los mensajes de consigna originales pero la autoría se perderá, a menos que seleccione *Por autor* o personalice el mensaje de consigna para incluirla.

The original commit messages can optionally be overridden with a custom commit message.

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

Add-on ID `weblate.json.customize`

Configuración

<code>sort_keys</code>	Ordenar claves de JSON	
<code>indent</code>	Sangría JSON	
<code>style</code>	Estilo de sangría de JSON	Available choices: spaces – Spaces tabs – Tabs

Triggers `storage post-load`

Permite ajustar el comportamiento de salida de JSON, por ejemplo, sangría o clasificación.

Formatea el archivo de propiedades de Java

Add-on ID `weblate.properties.sort`

Configuración *This add-on has no configuration.*

Triggers `repository pre-commit`

Ordena el archivo de propiedades de Java.

Retiro de comentarios obsoletos

Nuevo en la versión 3.7.

Add-on ID `weblate.removal.comments`

Configuración

<code>age</code>	Días para guardar	
------------------	-------------------	--

Triggers `daily`

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.

Add-on ID `weblate.removal.suggestions`

Configuración

<code>age</code>	Días para guardar	
<code>votes</code>	Umbral de votos	Umbral para eliminación. Este campo no surte ningún efecto si se desactivan los votos.

Triggers `daily`

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.

Add-on ID `weblate.resx.update`

Configuración *This add-on has no configuration.*

Triggers repository post-update

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.

Add-on ID `weblate.yaml.customize`

Configuración

in- dent	Sangría YAML	
width	Ajuste de renglones largos	Available choices: 80 – Wrap lines at 80 chars 100 – Wrap lines at 100 chars 120 – Wrap lines at 120 chars 180 – Wrap lines at 180 chars 65535 – No line wrapping
li- ne_ bre- ak	Saltos de renglón	Available choices: dos – DOS (\r\n) unix – UNIX (\n) mac – MAC (\r)

Triggers storage post-load

Permite ajustar el comportamiento de salida de YAML, como la longitud de los renglones y los saltos.

2.14.2 Customizing list of add-ons

The list of add-ons is configured by `WEBLATE_ADDONS`. To add another add-on, simply include the absolute class name in this setting.

2.14.3 Writing add-on

You can write your own add-ons too, create a subclass of `weblate.addons.base.BaseAddon` to define the add-on metadata, and then implement a callback to do the processing.

Ver también:

Developing add-ons

2.14.4 Executing scripts from add-on

Los add-ons pueden utilizarse para ejecutar scripts externos. Esto solía estar integrado en Weblate, pero ahora tienes que escribir algo de código para envolver tu script con un add-on.

```
#
# Copyright © 2012-2022 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."""

from django.utils.translation import gettext_lazy as _

from weblate.addons.events import EVENT_PRE_COMMIT
from weblate.addons.scripts import BaseScriptAddon

class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
    name = "weblate.example.pre"
    # Verbose name and long description
    verbose = _("Execute script before commit")
    description = _("This add-on executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{{ language_code }}.po"
```

Para obtener instrucciones de instalación, vea *Custom quality checks, add-ons and auto-fixes*.

El script se ejecuta con el directorio actual establecido en la raíz del repositorio VCS para cualquier componente dado.

Además, las variables de entorno siguientes están a su disposición:

WL_VCS

Sistema de control de versiones utilizado.

WL_REPO

URL del repositorio del proyecto originario.

WL_PATH

Ruta absoluta al repositorio del sistema de control de versiones.

WL_BRANCH

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

HEAD anterior después de la actualización (sólo disponible después de ejecutar el hook posterior a la actualización).

WL_COMPONENT_SLUG

Nuevo en la versión 3.9.

Componente slug utilizado para construir la URL.

WL_PROJECT_SLUG

Nuevo en la versión 3.9.

Slug del proyecto utilizado para construir la URL.

WL_COMPONENT_NAME

Nuevo en la versión 3.9.

Nombre del componente.

WL_PROJECT_NAME

Nuevo en la versión 3.9.

Nombre del proyecto.

WL_COMPONENT_URL

Nuevo en la versión 3.9.

URL del componente.

WL_ENGAGE_URL

Nuevo en la versión 3.9.

URL de participación del proyecto.

Ver también:

Configuración de componentes

Procesando el repositorio después de la actualización

Se puede utilizar para actualizar los archivos de traducción cuando el código fuente del VCS cambia. Para conseguirlo, recuerda que Weblate sólo ve los archivos confirmados en el VCS, por lo que debes commit los cambios como parte del script.

Por ejemplo, con Gulp puede hacerlo a través del código a continuación:

```
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Procesamiento de traducciones preconsigna

Utilice el script commitn para cambiar automáticamente una traducción antes de confirmarla en el repositorio.

Se pasa como un único parámetro que consiste en el nombre de archivo de una traducción actual.

2.15 Memoria de traducción

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* add-on.

For installation tips, see *Weblate Translation Memory*, which is turned on by default.

2.15.1 Ámbitos de la memoria de traducción

Nuevo en la versión 3.2: In earlier versions translation memory could be only loaded from a file corresponding to the current imported translation memory scope.

The translation memory scopes are there to allow both privacy and sharing of translations, to suit the desired behavior.

Memoria de traducción importada

Importing arbitrary translation memory data using the `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 4.12 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:

- Sign in. Deletes the account password, preventing the user from signing in without requesting a new password.
- Password reset. Prevents new e-mails from being sent, avoiding spamming users with too many password reset attempts.

El valor predeterminado es de 10.

Ver también:

Rate limiting

2.16.5 AUTO_UPDATE

Nuevo en la versión 3.2.

Distinto en la versión 3.11: 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* add-on
- *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:

Definiciones de idioma

2.16.13 BORG_EXTRA_ARGS

Nuevo en la versión 4.9.

You can pass additional arguments to **borg create** when built-in backups are triggered.

Ejemplo:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Ver también:

Respaldar y trasladar Weblate, borg create

2.16.14 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)

2.16.15 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: Cambiar esta opción afectará solo a las traducciones que se modifiquen en adelante; las existentes permanecerán almacenadas en la base de datos. Para aplicar los cambios también en las traducciones almacenadas, ejecute *updatechecks*.

Ver también:

Comprobaciones de calidad, Personalizar el comportamiento mediante indicadores

2.16.16 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.

2.16.17 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`

2.16.18 CONTACT_FORM

Nuevo en la versión 4.6.

Configures how e-mail from the contact form is being sent. Choose a configuration that matches 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.

2.16.19 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*.

The Docker container uses a separate volume for this, see *Volúmenes de contenedores Docker*.

media Default location for Django media files, specified by `MEDIA_ROOT`. Contains uploaded screenshots, see *Contexto visual para cadenas*.

vcs Version control repositories for translations.

backups Daily backup data, please check *Datos volcados para las copias de respaldo* for details.

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`, `Respaldar y trasladar Weblate`

2.16.20 DATABASE_BACKUP

Nuevo en la versión 3.1.

Define si las copias de respaldo de la base de datos deben almacenarse como texto sencillo, comprimidas u omitirse. Los valores autorizados son:

- `"plain"`
- `"compressed"`
- `"none"`

Ver también:

`Respaldar y trasladar Weblate`

2.16.21 DEFAULT_ACCESS_CONTROL

Nuevo en la versión 3.3.

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`

2.16.22 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`

2.16.23 DEFAULT_RESTRICTED_COMPONENT

Nuevo en la versión 4.1.

El valor predeterminado para la restricción de componentes.

Ver también:

Acceso restringido, Ámbito de los grupos

2.16.24 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, Commit, add, delete, merge, add-on, and merge request messages

2.16.25 DEFAULT_ADDONS

Default add-ons to install on every created component.

Nota: Esta configuración solo surte efecto en los componentes nuevos.

Ejemplo:

```
DEFAULT_ADDONS = {  
    # Add-on with no parameters  
    "weblate.flags.target_edit": {},  
    # Add-on with parameters  
    "weblate.autotranslate.autotranslate": {  
        "mode": "suggest",  
        "filter_type": "todo",  
        "auto_source": "mt",  
        "component": "",  
        "engines": ["weblate-translation-memory"],  
        "threshold": "80",  
    },  
}
```

Ver también:

install_addon, Complementos, WEBLATE_ADDONS

2.16.26 DEFAULT_COMMITER_EMAIL

Nuevo en la versión 2.4.

Committer e-mail address defaulting to noreply@weblate.org.

Ver también:

DEFAULT_COMMITER_NAME

2.16.27 DEFAULT_COMMITER_NAME

Nuevo en la versión 2.4.

El nombre del consignante, cuyo valor predeterminado es `Weblate`.

Ver también:

DEFAULT_COMMITER_EMAIL

2.16.28 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:

Definiciones de idioma, Idioma del código fuente

2.16.29 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

2.16.30 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*.

2.16.31 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

2.16.32 DEFAULT_PULL_MESSAGE

Configures the default title and message for pull requests.

2.16.33 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

2.16.34 ENABLE_HOOKS

Whether to enable anonymous remote hooks.

Ver también:

Actuadores de notificación

2.16.35 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*

2.16.36 ENABLE_SHARING

Turn on/off the *Share* menu so users can share translation progress on social networks.

2.16.37 GET_HELP_URL

Nuevo en la versión 4.5.2.

URL en el que pueden encontrarse recursos de ayuda para su instalación de Weblate.

2.16.38 GITEA_CREDENTIALS

Nuevo en la versión 4.12.

List for credentials for Gitea servers.

Consejo: Use this in case you want Weblate to interact with more of them, for single Gitea endpoint stick with *GITEA_USERNAME* and *GITEA_TOKEN*.

```
GITEA_CREDENTIALS = {
  "try.gitea.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitea.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

2.16.39 GITEA_USERNAME

Nuevo en la versión 4.12.

Gitea username used to send pull requests for translation updates.

Ver también:

GITEA_CREDENTIALS, *Gitea pull requests*

2.16.40 GITEA_TOKEN

Nuevo en la versión 4.12.

Gitea personal access token used to make API calls to send pull requests for translation updates.

Ver también:

GITEA_CREDENTIALS, *Gitea pull requests*, *Creating a Gitea personal access token*

2.16.41 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",
  },
}
```

(continué en la próxima página)

(proviene de la página anterior)

```
} ,  
}
```

2.16.42 GITLAB_USERNAME

GitLab username used to send merge requests for translation updates.

Ver también:

GITLAB_CREDENTIALS, *GitLab merge requests*

2.16.43 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 merge requests*, *GitLab: Personal access token*

2.16.44 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",  
  },  
}
```

2.16.45 GITHUB_USERNAME

GitHub username used to send pull requests for translation updates.

Ver también:

GITHUB_CREDENTIALS, *GitHub pull requests*

2.16.46 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 pull requests*, *Creating a GitHub personal access token*

2.16.47 GOOGLE_ANALYTICS_ID

Google Analytics ID to turn on monitoring of Weblate using Google Analytics.

2.16.48 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.

2.16.49 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.

2.16.50 INTERLEDGER_PAYMENT_POINTERS

Nuevo en la versión 4.12.1.

List of Interledger Payment Pointers (ILPs) for Web Monetization.

If multiple are specified, probabilistic revenue sharing is achieved by selecting one randomly.

Please check <<https://webmonetization.org/>> for more details.

Consejo: The default value lets users fund Weblate itself.

2.16.51 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

2.16.52 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

2.16.53 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

2.16.54 LEGAL_URL

Nuevo en la versión 3.5.

URL where your Weblate instance shows its legal documents.

Consejo: Useful if you host your legal documents outside Weblate for embedding them inside Weblate, please check *Información legal* for details.

Ejemplo:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Ver también:

PRIVACY_URL

2.16.55 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",
    ),
]
```

2.16.56 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

2.16.57 LICENSE_REQUIRED

Defines whether the license attribute in *Configuración de componentes* is required.

Nota: This is off by default.

2.16.58 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`.

2.16.59 LOCALIZE_CDN_URL y LOCALIZE_CDN_PATH

These settings configure the *CDN de regionalización de JavaScript* add-on. `LOCALIZE_CDN_URL` defines root URL where the localization CDN is available and `LOCALIZE_CDN_PATH` defines path where Weblate should store generated files which will be served at the `LOCALIZE_CDN_URL`.

Consejo: On Hosted Weblate, this uses `https://weblate-cdn.com/`.

Ver también:

CDN de regionalización de JavaScript

2.16.60 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`

2.16.61 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
)
```

2.16.62 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`

2.16.63 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`

2.16.64 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.

Consejo: When using Docker container, this configuration is automatically generated based on provided API keys, see *Configuración de traducción automática*.

```
MT_SERVICES = (  
    "weblate.machinery.apertium.ApertiumAPYTranslation",  
    "weblate.machinery.deepl.DeepLTranslation",  
    "weblate.machinery.glosbe.GlosbeTranslation",  
    "weblate.machinery.google.GoogleTranslation",  
    "weblate.machinery.libretranslate.LibreTranslateTranslation",  
    "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

2.16.65 MT_APERTIUM_APY

URL del servidor Apertium-APy, <https://wiki.apertium.org/wiki/Apertium-apy>

Ver también:

Apertium, Traducción automática, Sugerencias automáticas

2.16.66 MT_AWS_ACCESS_KEY_ID

Access key ID for Amazon Translate.

Ver también:

AWS, Traducción automática, Sugerencias automáticas

2.16.67 MT_AWS_SECRET_ACCESS_KEY

API secret key for Amazon Translate.

Ver también:

AWS, Traducción automática, Sugerencias automáticas

2.16.68 MT_AWS_REGION

Region name to use for Amazon Translate.

Ver también:

AWS, Traducción automática, Sugerencias automáticas

2.16.69 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

2.16.70 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

2.16.71 MT_DEEPL_API_URL

Distinto en la versión 4.7: The full API URL is now configured to allow using the free plan. Before, it was only possible to configure the API version using MT_DEEPL_API_VERSION.

API URL to use with the DeepL service. At the time of writing, there is the v1 API as well as a free and a paid version of the v2 API.

`https://api.deepl.com/v2/` (default in Weblate) Is meant for API usage on the paid plan, and the subscription is usage-based.

`https://api-free.deepl.com/v2/` Is meant for API usage on the free plan, and the subscription is usage-based.

`https://api.deepl.com/v1/` Is meant for CAT tools and is usable with a per-user subscription.

Previously Weblate was classified as a CAT tool by DeepL, so it was supposed to use the v1 API, but now is supposed to use the v2 API. Therefore it defaults to v2, and you can change it to v1 in case you have an existing CAT subscription and want Weblate to use that.

The easiest way to find out which one to use is to open an URL like the following in your browser:

https://api.deepl.com/v2/translate?text=Hello&target_lang=FR&auth_key=XXX

Replace the XXX with your auth_key. If you receive a JSON object which contains «Bonjour», you have the correct URL; if not, try the other three.

Ver también:

DeepL, Traducción automática, Sugerencias automáticas

2.16.72 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

2.16.73 MT_LIBRETRANSLATE_API_URL

Nuevo en la versión 4.7.1.

API URL for the LibreTranslate instance to use.

`https://libretranslate.com/` (official public instance) Requires an API key to use outside of the website.

Mirrors are documented on the LibreTranslate GitHub repository, some of which can be used without authentication:

<https://github.com/LibreTranslate/LibreTranslate#user-content-mirrors>

Ver también:

LibreTranslate, Traducción automática, Sugerencias automáticas

2.16.74 MT_LIBRETRANSLATE_KEY

Nuevo en la versión 4.7.1.

API key for the LibreTranslate instance specified in `MT_LIBRETRANSLATE_API_URL`.

Ver también:

LibreTranslate, Traducción automática, Sugerencias automáticas

2.16.75 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

2.16.76 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.

2.16.77 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

2.16.78 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)

2.16.79 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`.

2.16.80 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

2.16.81 MT_MICROSOFT_REGION

Region prefix as defined in the «Authenticating with a Multi-service resource» section.

2.16.82 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.

2.16.83 MT_MODERNMT_KEY

Clave de API para el motor de traducciones automáticas ModernMT.

Ver también:

ModernMT MT_MODERNMT_URL

2.16.84 MT_MODERNMT_URL

URL of ModernMT. It defaults to `https://api.modernmt.com/` for the cloud service.

Ver también:

ModernMT *MT_MODERNMT_KEY*

2.16.85 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*

2.16.86 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*

2.16.87 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*

2.16.88 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*

2.16.89 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*

2.16.90 MT_TMSERVER

URL where tmserver is running.

Ver también:

tmserver, *Traducción automática*, *Sugerencias automáticas*, *tmserver*

2.16.91 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

2.16.92 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

2.16.93 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

2.16.94 MT_SAP_BASE_URL

API URL to the SAP Translation Hub service.

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.95 MT_SAP_SANDBOX_APIKEY

API key for sandbox API usage

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.96 MT_SAP_USERNAME

Your SAP username

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.97 MT_SAP_PASSWORD

Your SAP password

Ver también:

SAP Translation Hub, Traducción automática, Sugerencias automáticas

2.16.98 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

2.16.99 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*.

2.16.100 DEFAULT_PAGE_LIMIT

Nuevo en la versión 4.7.

Default number of elements to display when pagination is active.

2.16.101 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",
    },
}
```

2.16.102 PAGURE_USERNAME

Nuevo en la versión 4.3.2.

Pagure username used to send merge requests for translation updates.

Ver también:

PAGURE_CREDENTIALS, Pagure merge requests

2.16.103 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 merge requests*, *Pagure API*

2.16.104 PRIVACY_URL

Nuevo en la versión 4.8.1.

URL where your Weblate instance shows its privacy policy.

Consejo: Useful if you host your legal documents outside Weblate for embedding them inside Weblate, please check *Información legal* for details.

Ejemplo:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Ver también:

LEGAL_URL

2.16.105 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*

2.16.106 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*

2.16.107 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`

2.16.108 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*

2.16.109 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.

2.16.110 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$"
```

2.16.111 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](#)

2.16.112 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](#)

2.16.113 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](#).

2.16.114 SENTRY_DSN

Nuevo en la versión 3.9.

Sentry DSN to use for *Collecting error reports*.

Ver también:

[Django integration for Sentry](#)

2.16.115 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](#)

2.16.116 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.

2.16.117 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`

2.16.118 SITE_TITLE

El título que se utilizará en el sitio web y los mensajes de correo que se envíen.

2.16.119 SPECIAL_CHARS

Caracteres adicionales que incluir en el teclado visual; *Visual keyboard*.

El valor predeterminado es:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "...")
```

2.16.120 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"
```

2.16.121 SSH_EXTRA_ARGS

Nuevo en la versión 4.9.

Allows to add custom parameters when Weblate is invoking SSH. This is useful when connecting to servers using legacy encryption or other non-standard features.

For example when SSH connection in Weblate fails with *Unable to negotiate with legacyhost: no matching key exchange method found. Their offer: diffie-hellman-group1-sha1*, you can enable that using:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"
```

Consejo: The string is evaluated by shell, so make sure to quote any whitespace and special characters.

Ver también:

[OpenSSH Legacy Options](#)

2.16.122 STATUS_URL

The URL where your Weblate instance reports its status.

2.16.123 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.

2.16.124 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:

Definiciones de idioma incorporadas

2.16.125 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.

2.16.126 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

2.16.127 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
```

2.16.128 WEBLATE_ADDONS

List of add-ons available for use. To use them, they have to be enabled for a given translation component. By default this includes all built-in add-ons, when extending the list you will probably want to keep existing ones enabled, for example:

```
WEBLATE_ADDONS = (  
    # Built-in add-ons  
    "weblate.addons.gettext.GenerateMoAddon",  
    "weblate.addons.gettext.UpdateLinguasAddon",  
    "weblate.addons.gettext.UpdateConfigureAddon",  
    "weblate.addons.gettext.MsgmergeAddon",  
    "weblate.addons.gettext.GettextCustomizeAddon",  
    "weblate.addons.gettext.GettextAuthorComments",  
)
```

(continué en la próxima página)

(proviene de la página anterior)

```
"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",
# Add-on you want to include
"weblate.addons.example.ExampleAddon",
)
```

Nota: Removing the add-on from the list does not uninstall it from the components. Weblate will crash in that case. Please uninstall add-on from all components prior to removing it from this list.

Ver también:

Complementos, DEFAULT_ADDONS

2.16.129 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

2.16.130 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

2.16.131 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

2.16.132 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 © 2012–2022 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

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

#
# Django settings for Weblate project.
```

(continué en la próxima página)

(proviene de la página anterior)

```

#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # Uncomment for MySQL older than 5.7:
            # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
            # Set emoji capable charset for MySQL:
            # "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away error:
            # "connect_timeout": 28800,
        },
        # Persistent connections
        "CONN_MAX_AGE": 0,
        # Disable server-side cursors, might be needed with pgbouncer
        "DISABLE_SERVER_SIDE_CURSORS": False,
    }
}

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"

```

(continué en la próxima página)

(proviene de la página anterior)

```

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "ไทย"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "简体中文"),
    ("zh-hant", "繁體中文"),
)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# 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

```

(continué en la próxima página)

(proviene de la página anterior)

```

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(DATA_DIR, "static")

# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# You can generate it using weblate/examples/generate-secret-key
SECRET_KEY = ""

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
        },
        "APP_DIRS": True,
    }
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.

```

(continué en la próxima página)

(proviene de la página anterior)

```

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.BitbucketOAuth2",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_GITHUB_ORG_KEY = ""
SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""

SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
    "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",

```

(continué en la próxima página)

(proviene de la página anterior)

```

"weblate.accounts.pipeline.ensure_valid",
"weblate.accounts.pipeline.remove_account",
"social_core.pipeline.social_auth.associate_by_email",
"weblate.accounts.pipeline.reauthenticate",
"weblate.accounts.pipeline.verify_username",
"social_core.pipeline.user.create_user",
"social_core.pipeline.social_auth.associate_user",
"social_core.pipeline.social_auth.load_extra_data",
"weblate.accounts.pipeline.cleanup_next",
"weblate.accounts.pipeline.user_full_name",
"weblate.accounts.pipeline.store_email",
"weblate.accounts.pipeline.notify_connect",
"weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/#account
↪"
SOCIAL_AUTH_PROTECTED_USER_FIELDS = ("email",)
SOCIAL_AUTH_SLUGIFY_USERNAMES = True
SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

# Password validation configuration
AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator" # noqa: E501, pylint: disable=line-too-long
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {"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")
    #     }
    # }

```

(continué en la próxima página)

(proviene de la página anterior)

```

#     }
# },
]

# 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",
    "weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",

```

(continué en la próxima página)

(proviene de la página anterior)

```

# 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

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"},

```

(continué en la próxima página)

(proviene de la página anterior)

```

    "django.server": {
        "(): "django.utils.log.ServerFormatter",
        "format": "[% (server_time)s] %(message)s",
    },
},
"handlers": {
    "mail_admins": {
        "level": "ERROR",
        "filters": ["require_debug_false"],
        "class": "django.utils.log.AdminEmailHandler",
        "include_html": True,
    },
    "console": {
        "level": "DEBUG",
        "class": "logging.StreamHandler",
        "formatter": "simple",
    },
    "django.server": {
        "level": "INFO",
        "class": "logging.StreamHandler",
        "formatter": "django.server",
    },
    "syslog": {
        "level": "DEBUG",
        "class": "logging.handlers.SysLogHandler",
        "formatter": "syslog",
        "address": "/dev/log",
        "facility": SysLogHandler.LOG_LOCAL2,
    },
    # Logging to a file
    # "logfile": {
    #     "level": "DEBUG",
    #     "class": "logging.handlers.RotatingFileHandler",
    #     "filename": "/var/log/weblate/weblate.log",
    #     "maxBytes": 100000,
    #     "backupCount": 3,
    #     "formatter": "logfile",
    # },
},
"loggers": {
    "django.request": {
        "handlers": ["mail_admins", DEFAULT_LOG],
        "level": "ERROR",
        "propagate": True,
    },
    "django.server": {
        "handlers": ["django.server"],
        "level": "INFO",
        "propagate": False,
    },
    # 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

```

(continué en la próxima página)

(proviene de la página anterior)

```

    "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # SAML IdP
    "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
},
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    # "weblate.machinery.apertium.ApertiumAPYTranslation",
    # "weblate.machinery.baidu.BaiduTranslation",
    # "weblate.machinery.deepl.DeepLTranslation",
    # "weblate.machinery.glosbe.GlosbeTranslation",
    # "weblate.machinery.google.GoogleTranslation",
    # "weblate.machinery.googlev3.GoogleV3Translation",
    # "weblate.machinery.libretranslate.LibreTranslateTranslation",
    # "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",
    # "weblate.machinery.netease.NeteaseSightTranslation",
    # "weblate.machinery.tmserver.AmagamaTranslation",
    # "weblate.machinery.tmserver.TMServerTranslation",
    # "weblate.machinery.yandex.YandexTranslation",
    # "weblate.machinery.saptranslationhub.SAPTranslationHub",
    # "weblate.machinery.youdao.YoudaoTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APY server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# 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

```

(continué en la próxima página)

(proviene de la página anterior)

```

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_Baidu_ID = None
MT_Baidu_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.
# ↪html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
# 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

```

(continué en la próxima página)

(proviene de la página anterior)

```

# Apply session cookie settings to language cookie as well
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# 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",

```

(continué en la próxima página)

(proviene de la página anterior)

```

# "weblate.checks.chars.BeginSpaceCheck",
# "weblate.checks.chars.EndSpaceCheck",
# "weblate.checks.chars.DoubleSpaceCheck",
# "weblate.checks.chars.EndStopCheck",
# "weblate.checks.chars.EndColonCheck",
# "weblate.checks.chars.EndQuestionCheck",
# "weblate.checks.chars.EndExclamationCheck",
# "weblate.checks.chars.EndEllipsisCheck",
# "weblate.checks.chars.EndSemicolonCheck",
# "weblate.checks.chars.MaxLengthCheck",
# "weblate.checks.chars.KashidaCheck",
# "weblate.checks.chars.PunctuationSpacingCheck",
# "weblate.checks.format.PythonFormatCheck",
# "weblate.checks.format.PythonBraceFormatCheck",
# "weblate.checks.format.PHPFormatCheck",
# "weblate.checks.format.CFormatCheck",
# "weblate.checks.format.PerlFormatCheck",
# "weblate.checks.format.JavaScriptFormatCheck",
# "weblate.checks.format.LuaFormatCheck",
# "weblate.checks.format.ObjectPascalFormatCheck",
# "weblate.checks.format.SchemeFormatCheck",
# "weblate.checks.format.CSharpFormatCheck",
# "weblate.checks.format.JavaFormatCheck",
# "weblate.checks.format.JavaMessageFormatCheck",
# "weblate.checks.format.PercentPlaceholdersCheck",
# "weblate.checks.format.VueFormattingCheck",
# "weblate.checks.format.I18NextInterpolationCheck",
# "weblate.checks.format.ESTemplateLiteralsCheck",
# "weblate.checks.angularjs.AngularJSInterpolationCheck",
# "weblate.checks.icu.ICUMessageFormatCheck",
# "weblate.checks.icu.ICUSourceCheck",
# "weblate.checks.qt.QtFormatCheck",
# "weblate.checks.qt.QtPluralCheck",
# "weblate.checks.ruby.RubyFormatCheck",
# "weblate.checks.consistency.PluralsCheck",
# "weblate.checks.consistency.SamePluralsCheck",
# "weblate.checks.consistency.ConsistencyCheck",
# "weblate.checks.consistency.TranslatedCheck",
# "weblate.checks.chars.EscapedNewlineCountingCheck",
# "weblate.checks.chars.NewLineCountCheck",
# "weblate.checks.markup.BBCodeCheck",
# "weblate.checks.chars.ZeroWidthSpaceCheck",
# "weblate.checks.render.MaxSizeCheck",
# "weblate.checks.markup.XMLValidityCheck",
# "weblate.checks.markup.XMLTagsCheck",
# "weblate.checks.markup.MarkdownRefLinkCheck",
# "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",
# )

```

(continué en la próxima página)

(proviene de la página anterior)

```

# List of automatic fixups
# AUTOFIX_LIST = (
#     "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
#     "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
#     "weblate.trans.autofixes.chars.RemoveZeroSpace",
#     "weblate.trans.autofixes.chars.RemoveControlChars",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LangaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",
#     "weblate.addons.generate.GenerateFileAddon",
#     "weblate.addons.generate.PseudolocaleAddon",
#     "weblate.addons.generate.PrefillAddon",
#     "weblate.addons.json.JSONCustomizeAddon",
#     "weblate.addons.properties.PropertiesSortAddon",
#     "weblate.addons.git.GitSquashAddon",
#     "weblate.addons.removal.RemoveComments",
#     "weblate.addons.removal.RemoveSuggestions",
#     "weblate.addons.resx.ResxUpdateAddon",
#     "weblate.addons.yaml.YAMLCustomizeAddon",
#     "weblate.addons.cdn.CDNJSAddon",
# )

# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

# List of URLs your site is supposed to serve
ALLOWED_HOSTS = ["*"]

# Configuration for caching
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # "LOCATION": "unix:///var/run/redis/redis.sock?db=1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
            # If you set password here, adjust CELERY_BROKER_URL as well
            "PASSWORD": None,
        }
    }
}

```

(continué en la próxima página)

(proviene de la página anterior)

```

        "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",
    "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 = (

```

(continué en la próxima página)

(proviene de la página anterior)

```

# rf"{URL_PREFIX}/accounts/(.*)$", # Required for login
# rf"{URL_PREFIX}/admin/login/(.*)$", # Required for admin login
# rf"{URL_PREFIX}/static/(.*)$", # Required for development mode
# rf"{URL_PREFIX}/widgets/(.*)$", # Allowing public access to widgets
# rf"{URL_PREFIX}/data/(.*)$", # Allowing public access to data exports
# rf"{URL_PREFIX}/hooks/(.*)$", # Allowing public access to notification hooks
# rf"{URL_PREFIX}/healthz/$", # Allowing public access to health check
# rf"{URL_PREFIX}/api/(.*)$", # Allowing access to API
# rf"{URL_PREFIX}/js/i18n/$", # JavaScript localization
# rf"{URL_PREFIX}/contact/$", # Optional for contact form
# rf"{URL_PREFIX}/legal/(.*)$", # Optional for legal app
# rf"{URL_PREFIX}/avatar/(.*)$", # Optional for avatars
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True
# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_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"},
    "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 = SITE_DOMAIN
AKISMET_API_KEY = None

```

2.18 Órdenes de gestión

Nota: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check *Permisos del sistema de archivos* for more details.

You will find basic management commands (available as `./manage.py` in the Django sources, or as an extended set in a script called **weblate** installable atop Weblate).

2.18.1 Invocar órdenes de gestión

Como ya se mencionó, la invocación dependerá de cómo instaló Weblate.

If using virtualenv for Weblate, you can either specify the full path to **weblate**, or activate the virtualenv prior to invoking it:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

If you are using source code directly (either from a tarball or Git checkout), the management script is `./manage.py` available in the Weblate sources. To run it:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip 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 cleanup_ssh_keys

weblate cleanup_ssh_keys

Nuevo en la versión 4.9.1.

Performs cleanup of stored SSH host keys:

- Removes deprecated RSA keys for GitHub which might cause issues connecting to GitHub.
- Removes duplicate entries in host keys.

Ver también:

Repositorios SSH

2.18.10 createadmin

weblate createadmin

Creates an `admin` account with a random password, unless it is specified.

--password PASSWORD

Permite proporcionar una contraseña mediante la línea de órdenes, de modo que no se genere una aleatoria.

--no-password

Do not set password, this can be useful with `-update`.

--username USERNAME

Use the given name instead of `admin`.

--email USER@EXAMPLE.COM

Specify the admin e-mail address.

--name

Specify the admin name (visible).

--update

Update the existing user (you can use this to change passwords).

Distinto en la versión 2.9: Added parameters `--username`, `--email`, `--name` and `--update`.

2.18.11 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

2.18.12 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.

2.18.13 import_demo

weblate import_demo

Nuevo en la versión 4.1.

Creates a demo project with components based on <<https://github.com/WeblateOrg/demo>>.

Puede ser de utilidad si está desarrollando Weblate.

2.18.14 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",
```

(continué en la próxima página)

(proviene de la página anterior)

```

    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]

```

Ver también:*import_memory*

2.18.15 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*

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>Distinto en la versión 3.0: The `import_project` command is now based on the *Detección de componentes* add-on, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on the file mask.

<project> names an existing project, into which the components are to be imported.

The <gitrepo> defines the Git repository URL to use, and <branch> signifies the Git branch. To import additional translation components from an existing Weblate component, use a `weblate://<project>/<component>` URL for the <gitrepo>.

The <filemask> defines file discovery for the repository. It can be either be made simple using wildcards, or it can use the full power of regular expressions.

The simple matching uses `**` for component name and `*` for language, for example: `**/* .po`The regular expression has to contain groups named *component* and *language*. For example: `(?P<language>[^/]*) / (?P<component>[^-/*]*) \.po`

The import matches existing components based on files and adds the ones that do not exist. It does not change already existing ones.

--name-template TEMPLATE

Customize the name of a component using Django template syntax.

For example: `Documentation: {{ component }}`

--base-file-template TEMPLATE

Customize the base file for monolingual translations.

For example: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Customize the base file for addition of new translations.

For example: `{{ component }}/ts/en.ts`

--file-format FORMAT

You can also specify the file format to use (see *Formatos de archivo admitidos*), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see *Configuración de componentes*) with this parameter. It has to be a valid regular expression.

--main-component

You can specify which component will be chosen as the main one—the one actually containing the VCS repository.

--license NAME

Specify the overall, project or component translation license.

--license-url URL

Specify the URL where the translation license is to be found.

--vcs NAME

In case you need to specify which version control system to use, you can do it here. The default version control is Git.

To give you some examples, let's try importing two projects.

First The Debian Handbook translations, where each language has separate a folder with the translations of each chapter:

```
weblate import_project \  
  debian-handbook \  
  git://anonscm.debian.org/debian-handbook/debian-handbook.git \  
  squeeze/master \  
  '*/**.po'
```

Then the Tanaguru tool, where the file format needs be specified, along with the base file template, and how all components and translations are located in single folder:

```
weblate import_project \  
  --file-format=properties \  
  --base-file-template=web-app/tgol-web-app/src/main/resources/i18n/%s-I18N.  
↪properties \  
  tanaguru \  
  https://github.com/Tanaguru/Tanaguru \  
  master \  
  web-app/tgol-web-app/src/main/resources/i18n/**-I18N_*.properties
```

More complex example of parsing of filenames to get the correct component and language out of a filename like `src/security/Numerous_security_holes_in_0.10.1.de.po`:

```
weblate import_project \  
  tails \  
  git://git.tails.boum.org/tails master \  
  'wiki/src/security/(?P<component>.*)(?P<language>[^\.]*)\.po$'
```

Filtering only translations in a chosen language:

```
./manage import_project \
  --language-regex '(cs|sk)$' \
  weblate \
  https://github.com/WeblateOrg/weblate.git \
  'weblate/locale/*/LC_MESSAGES/**/*.po'
```

Importing Sphinx documentation split to multiple files:

```
$ weblate import_project --name-template 'Documentation: %s' \
  --file-format po \
  project https://github.com/project/docs.git master \
  'docs/locale/*/LC_MESSAGES/**/*.po'
```

Importing Sphinx documentation split to multiple files and directories:

```
$ weblate import_project --name-template 'Directory 1: %s' \
  --file-format po \
  project https://github.com/project/docs.git master \
  'docs/locale/*/LC_MESSAGES/dir1/**/*.po'
$ weblate import_project --name-template 'Directory 2: %s' \
  --file-format po \
  project https://github.com/project/docs.git master \
  'docs/locale/*/LC_MESSAGES/dir2/**/*.po'
```

Ver también:

More detailed examples can be found in the starting chapter, alternatively you might want to use *import_json*.

2.18.17 importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by *dumpuserdata*

2.18.18 importusers

weblate importusers --check <file.json>

Imports users from JSON dump of the Django auth_users database.

--check

With this option it will just check whether a given file can be imported and report possible conflicts arising from usernames or e-mails.

You can dump users from the existing Django installation using:

```
weblate dumpdata auth.User > users.json
```

2.18.19 install_addon

Nuevo en la versión 3.2.

weblate install_addon --addon ADDON <project|project/component>

Installs an add-on to a set of components.

--addon ADDON

Name of the add-on to install. For example `weblate.gettext.customize`.

--configuration CONFIG

JSON encoded configuration of an add-on.

--update

Update the existing add-on configuration.

You can either define which project or component to install the add-on in (for example `weblate/application`), or use `--all` to include all existing components.

To install *Personalizar la salida de gettext* for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --  
↪update --all
```

Ver también:

Complementos

2.18.20 list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names.

This is used to generate `<https://wiki.110n.cz/Slovn%C3%ADk_s_n%C3%A1lvy_jazyk%C5%AF>`.

2.18.21 list_translators

weblate list_translators <project|project/component>

Lists translators by contributed language for the given project:

```
[French]  
Jean Dupont <jean.dupont@example.com>  
[English]  
John Doe <jd@example.com>
```

--language-code

List names by language code instead of language name.

You can either define which project or component to use (for example `weblate/application`), or use `--all` to list translators from all existing components.

2.18.22 list_versions

weblate list_versions

Lists all Weblate dependencies and their versions.

2.18.23 loadpo

weblate loadpo <project|project/component>

Reloads translations from disk (for example in case you have done some updates in the VCS repository).

--force

Force update, even if the files should be up-to-date.

--lang LANGUAGE

Limit processing to a single language.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

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.

2.18.24 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`

2.18.25 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.

2.18.26 pushgit

weblate pushgit <project|project/component>

Pushes committed changes to the upstream VCS repository.

--force-commit

Force commits any pending changes, prior to pushing.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Nota: Weblate pushes changes automatically if *Enviar al consignar* in *Configuración de componentes* is turned on, which is the default.

2.18.27 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

2.18.28 setupgroups

weblate setupgroups

Configura los grupos predeterminados y, opcionalmente, asigna todos los usuarios a ese grupo predeterminado.

--no-privs-update

Desactiva la actualización automática de los grupos existentes (solamente añade los nuevos).

--no-projects-update

Prevents automatic updates of groups for existing projects. This allows adding newly added groups to existing projects, see *Control de acceso al proyecto*.

Ver también:

Lista de privilegios y roles incorporados

2.18.29 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).

2.18.30 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.

2.18.31 updategit

weblate updategit <project|project/component>

Fetches remote VCS repositories and updates the internal cache.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

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».


Proporciona información a tus traductores publicando anuncios, en todo el sitio, por proyecto, componente o idioma.

Anunciar el propósito, los plazos, el estado, o especificar los objetivos de la traducción.

Los usuarios recibirán una notificación por cada anuncio en los proyectos que monitorizan (salvo que decidan no recibirlas).

Esto puede ser útil para varias cosas, desde anunciar el propósito del sitio web hasta especificar los objetivos de las traducciones.

Se pueden publicar los anuncios en cada nivel en el menú *Manage*, utilizando *Post announcement*:






 Weblate


Dashboard

Projects ▾

Languages ▾

Checks ▾



 WeblateOrg

translated 90%

Translations will be used only if they reach 60%. 

Components

Languages

Info

Search

Insights ▾

Files ▾

Tools ▾

Manage ▾

Share ▾

 Not watching ▾

Post announcement 

Message

You can use Markdown and mention users by @username.

Category

Info (light blue) ▾

Category defines color used for the message.

Expiry date

mm/dd/yyyy 

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ Notify users

The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.

Add

Powered by Weblate 4.12 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

También se puede añadir mediante la interfaz de administración:

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

A continuación, los anuncios se muestran en función de su contexto especificado:

Ningún contexto especificado

Se muestra en el panel de herramientas (página de destino).

Proyecto especificado

Se muestra dentro del proyecto, incluyendo todos sus componentes y traducciones.

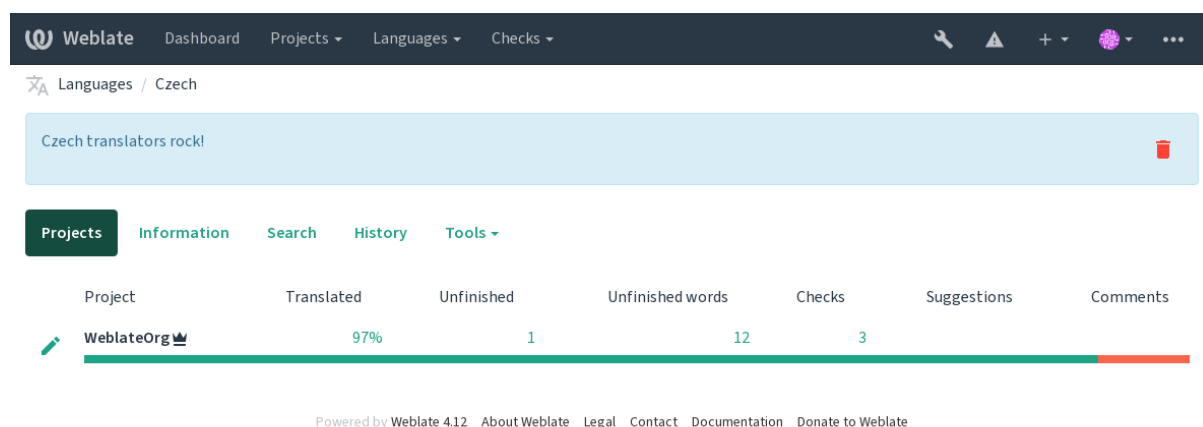
Componente especificado

Se muestra para un componente determinado y todas sus traducciones.

Idioma especificado

Se muestra en el resumen de idiomas y en todas las traducciones en ese idioma.

Así es como se ve en la página de resumen de idiomas:



2.20 Listas de componentes

Especifica varias listas de componentes para que aparezcan como opciones en el panel de control del usuario, de las cuales los usuarios pueden elegir una como vista predeterminada. Para saber más, ver [Cuadro de mando](#).

Distinto en la versión 2.20: Se presentará un estado para cada lista de componentes presentada en el tablero.

Los nombres y el contenido de las listas de componentes pueden especificarse en la interfaz de administración, en la sección: *Component lists*. Cada lista de componentes debe tener un nombre que se muestre al usuario, y un slug que lo represente en la URL.

Distinto en la versión 2.13: Cambia la configuración del panel de control para los usuarios anónimos desde la interfaz de administración, modificando el panel de control que se presenta a los usuarios no autenticados.

2.20.1 Listas de componentes automáticas

Nuevo en la versión 2.13.

Añadir componentes a la lista automáticamente en función de su slug creando reglas: *Automatic component list assignment*.

- Es útil para mantener listas de componentes para instalaciones grandes, o en caso de que quieras tener una lista de componentes con todos los componentes de tu instalación de Weblate.

Consejo: Haz una lista de componentes que contenga todos los componentes de tu instalación de Weblate.

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:

Weblate administration
WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Component lists · Add Component list

Add Component list

Required fields are marked in bold.

Component list name:
Display name

URL slug:
Name used in URLs and filenames.

☒ Show on dashboard
When enabled this component list will be shown as a tab on the dashboard

Components:

Available components ⓘ

- WeblateOrg/Django
- WeblateOrg/Language names
- WeblateOrg/WeblateOrg

Chosen components ⓘ

Choose all ⓘ Remove all ⓘ

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE? ⓘ
<input type="text" value="^.*\$"/>	<input type="text" value="^.*\$"/>	<input type="button" value="✕"/>

[+ Add another Automatic component list assignment](#)

2.21 Módulos opcionales de Weblate

Hay varios módulos opcionales disponibles para que los configure.

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 6F93F20AD23CE7A7B2A5848DAE344EA6B5336AC3, for which the corresponding public key is found below.

```

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNB8Jf0zsBDADOLT7QN1aH7Oud/vj/XKAReLa/2eYLPbFNYopoQAKaUzOaVQ7a
09MviRbMAFGVGPZ+XR2o8UPm3+XbXhB2NzFWgIUDhFE3A3vqho/4cu2auBDKWIOs
dJ3S3aT3qFe/TYpJhDgv64nwNZ7Z5YAOmZLcC+RCGfBhBONvRxjjJa4mXmxd2G4
Q8F9qrHyvTH5G8jE4zOCi8ghleJhfzcPLqzetStyT6jkl7ZnIUkfAIktoLqsN
qsi3kjmknkCQ84DKwOBjG0SmhYFzZmAeWkYOIDDsOcDUJv09cc2GmNhr3jHUpY2B
I6U3l1j6QRyMWXbcEIIIBYbxM5oE/nL7poZnzUfjyR2q0UBzJb9zN+gTcPjfrmvH
Xr1Or+QrlySUn3A3wNjNGEKGwG9ceAq6419ixHMr9nSxnNC3GT72avJT5DFd/jha
xrR626glO7TR9zlg276SbIsLCE/EkASql8A5j2pDizEkfREImTb27megRE3977i
0BEYod27U+Sdsw8AEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlMmNvbT6J
Ac4EEwEKADgWIQRvk/IK0jznp7KIhl2uNE6mtTNqwwUCYI/TOWIbAwULCQgHAgYV
CgklCwIEFgIDAQIeAQIXgAAKCRcuNE6mtTNqww7T8DADJKNLLFiN0KQO+FPSSDBLH
k8esr4c3IH5URgKw2ML7uzAnQ+Hy1rxxVYAIUlr0CMcfou2zuFlsTlrjQ057k8Z
-----END PGP PUBLIC KEY BLOCK-----

```

Powered by Weblate 4.12
 [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
Creating new project	PROJECT	5	600	600

If a user fails to sign in `AUTH_LOCK_ATTEMPTS` times, password authentication will be turned off on the account until having gone through the process of having its password reset.

The settings can be also applied in the Docker container by adding `WEBLATE_` prefix to the setting name, for example `RATELIMIT_ATTEMPTS` becomes `WEBLATE_RATELIMIT_ATTEMPTS`.

The API has separate rate limiting settings, see [API 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:

How to manage static files (e.g. images, JavaScript, CSS), *Serving static files*

2.22.3 Custom quality checks, add-ons and auto-fixes

To install your code for *Correcciones automáticas personalizadas*, *Escribir los propios controles* or *Writing add-on* in Weblate:

1. Place the files into your Python module containing the Weblate customization (see *Crear un módulo Python*).
2. Add its fully-qualified path to the Python class in the dedicated settings (`WEBLATE_ADDONS`, `CHECK_LIST` or `AUTOFIX_LIST`):

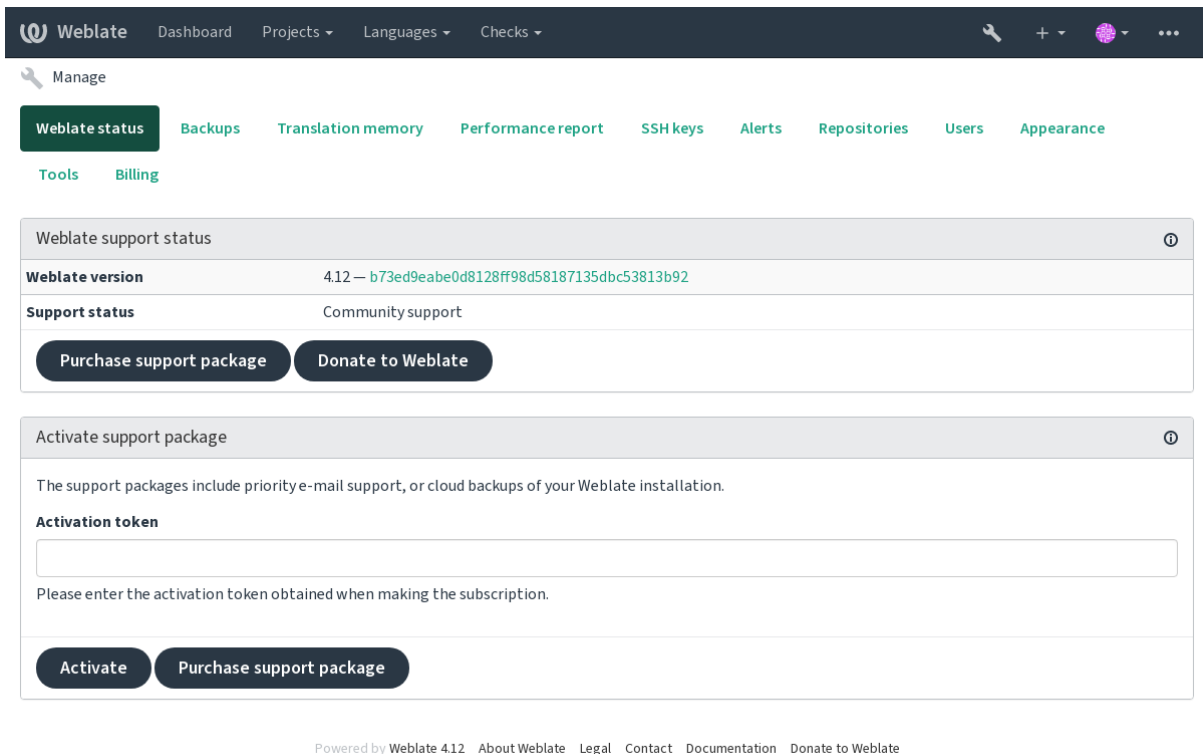
```
# Checks  
CHECK_LIST += ("weblate_customization.checks.FooCheck",)  
  
# Autofixes  
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)  
  
# Add-ons  
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Ver también:

Correcciones automáticas personalizadas, *Escribir los propios controles*, *Writing add-on*, *Executing scripts from add-on*

2.23 Interfaz de gestión

La interfaz de gestión ofrece ajustes de administración bajo la URL `/manage/`. Está disponible para los usuarios que hayan iniciado sesión con privilegios de administrador, accediendo a través del icono de la llave inglesa en la parte superior derecha:



The screenshot shows the Weblate management interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with a grid of buttons: Weblate status (highlighted), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, and Appearance. Underneath are Tools and Billing links.

The 'Weblate status' section contains the following information:

- Weblate support status** (with an info icon)
- Weblate version**: 4.12 — b73ed9eabe0d8128ff98d58187135dbc53813b92
- Support status**: Community support
- Buttons: **Purchase support package** and **Donate to Weblate**

Below this is the 'Activate support package' section (with an info icon):

- Text: The support packages include priority e-mail support, or cloud backups of your Weblate installation.
- Activation token**: A text input field.
- Text: Please enter the activation token obtained when making the subscription.
- Buttons: **Activate** and **Purchase support package**

At the bottom, a footer line reads: Powered by Weblate 4.12 About Weblate Legal Contact Documentation Donate to Weblate

Incluye un resumen básico de tu Weblate:

- Estado de servicio técnico; vea *Obtener ayuda con Weblate*
- Copias de respaldo; vea *Respaldar y trasladar Weblate*
- Memoria de traducción compartida; vea *Memoria de traducción*
- Informe de rendimiento para revisar el funcionamiento de Weblate y la duración de las consultas de Celery
- Gestión de claves SSH; vea *Repositorios SSH*
- Resumen de alertas para todos los componentes, ver alerts.

2.23.1 La interfaz administrativa de Django

Advertencia: Se eliminará en el futuro, ya que se desaconseja su uso; la mayoría de las funciones se pueden gestionar directamente en Weblate.

Aquí puede gestionar los objetos almacenados en la base de datos, tales como usuarios, traducciones y determinadas configuraciones:

Webplate administration

WELCOME, **WEBPLATE TEST** / [RETURN TO WEBPLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Site administration

REPORTS

Weblate support status

Status of repositories

SSH keys

Performance report

Translation memory

ACCOUNTS

Audit log entries

User profiles

Verified e-mails

+ Add

Change

+ Add

Change

+ Add

Change

AUTH TOKEN

Tokens

+ Add

Change

AUTHENTICATION

Groups

Roles

Users

+ Add

Change

+ Add

Change

+ Add

Change

BILLING

Billing plans

Customer billings

Invoices

+ Add

Change

+ Add

Change

+ Add

Change

FONTS

Font groups

Fonts

+ Add

Change

+ Add

Change

LEGAL

TOS agreements

+ Add

Change

PYTHON SOCIAL AUTH

Associations

Nonces

User social auths

+ Add

Change

+ Add

Change

+ Add

Change

SCREENSHOTS

Screenshots

+ Add

Change

TRANSLATION MEMORY

Translation memory entries

+ Add

Change

WEBPLATE CONFIGURATION

Settings

+ Add

Change

WEBPLATE LANGUAGES

Languages

+ Add

Change

WEBPLATE TRANSLATIONS

Announcements

Component lists

Components

Contributor agreements

Projects

+ Add

Change

+ Add

Change

+ Add

Change

+ Add

Change

+ Add

Change

Recent actions

My actions

None available

En la sección *Reports*, puedes comprobar el estado de tu sitio, ajústalo para *Puesta en marcha de entorno de producción*, o gestiona las claves SSH utilizadas para acceder a *Accessing repositories*.

Gestiona los objetos de la base de datos en cualquiera de las secciones. La más interesante es probablemente *Weblate translations*, donde puedes gestionar proyectos traducibles, ver *Configuración de proyectos* y *Configuración de componentes*.

Weblate languages contiene las definiciones de los idiomas, que se explican en *Definiciones de idioma*.

Añadir un proyecto

Añadir un proyecto sirve de contenedor para todos los componentes. Normalmente se crea un proyecto para una pieza de software, o libro (Ver *Configuración de proyectos* para información sobre los parámetros individuales):

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:
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

Ver también:

Configuración de proyectos

Componentes bilingües

Una vez añadido un proyecto, se pueden añadir componentes de traducción al mismo. (Véase *Configuración de componentes* para obtener información sobre los parámetros individuales):

[illegible]

Ver también:

Configuración de componentes, Formatos bilingües y monolingües

Componentes monolingües

Para facilitar la traducción de los mismos, proporciona un archivo de plantilla que contenga la asignación de los ID de los mensajes a su respectivo idioma de origen (normalmente el inglés). (Véase *Configuración de componentes* para obtener información sobre los parámetros individuales):

Weblate administration
[Help](#)
[Weblate 1.0.0](#)
[Return to Weblate](#)
[Documentation](#)
[Feedback](#)
[Feedback](#)

[Home](#)
[Recent translations](#)
[Components](#)
[Add component](#)

Add Component

Required fields are marked with an asterisk (*)

Component name

URL slug

Project

Weblate

Version control system

Git

Source code repository

Repository push URL

Repository branch

Expected repository URL

Source string file reporting address

Repository branch

Push branch

File mask

Metadata base language file

Use base file

☒

Intermediate language file

Language for new translations

File format

☐ Locked

☒ Allow translation propagation

☒ Use on all suggestions

☐ Suggestion voting

Automatically suggest updates with this number of votes, use 0 to turn it off

Translation flags

Enforced checks

Translation branches

Contributor agreement

Add new translation

Language code style

☐ Manage strings

☒ Merge strings

Merge style

Control message when translating

Control message when adding translation

Control message when merging translation

Control message when merging translation

Control message when add-on makes a change

Merge request message

☒ Push on commit

☒ Push on merge

Age of changes to commit

☒ Push on merge

Source language

Language filter

Priority

☐ Restricted component

Shows in projects

☐ Use as a gallery

Privacy policy

Localisation

Save and add another

Save and update details

Save

2.23. Interfaz de gestión

397

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.

Manage

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

Tools Billing

Weblate support status ⓘ

Weblate version 4.12 — b73ed9eabe0d8128ff98d58187135dbc53813b92

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 4.12 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:

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	4.12 — b73ed9eabe0d8128ff98d58187135dbc53813b92
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

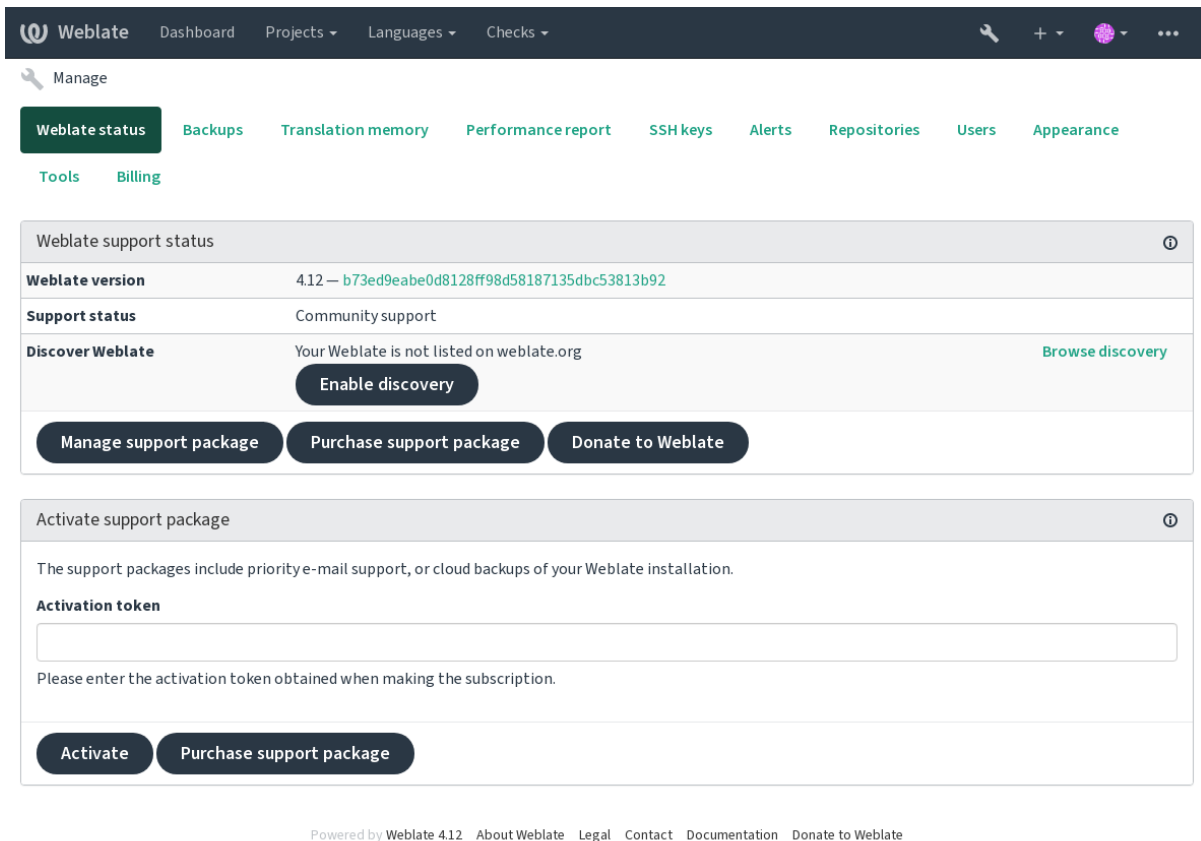
Please enter the activation token obtained when making the subscription.

[Activate](#)
[Purchase support package](#)

 Powered by Weblate 4.12
 [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:



The screenshot shows the Weblate dashboard interface. At the top, there's a navigation bar with 'Weblate' logo and links to Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with various tabs: Weblate status (active), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, and Appearance. Under 'Weblate status', there are sub-tabs for Tools and Billing. The main content area displays the 'Weblate support status' section, which includes the current version (4.12), support status (Community support), and a 'Discover Weblate' section with an 'Enable discovery' button. At the bottom of this section are buttons for 'Manage support package', 'Purchase support package', and 'Donate to Weblate'. Below this is the 'Activate support package' section, which includes a text box for an activation token and buttons for 'Activate' and 'Purchase support package'.

Powered by Weblate 4.12 [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*
- *Contributing to Weblate modules*
- *Traducir Weblate*
- *Contribute to Weblate documentation*
- *Weblate discussions*
- *Funding Weblate development*

3.1.1 Traducir Weblate

Weblate is continually being [translated](#) using Weblate itself. Feel free to take your part in the effort of making Weblate available in as many human languages as possible. It brings Weblate closer to its users!

If you find a possible mistake in the source string, you can mark it with a comment in the Weblate editor. This way, it can be discussed and corrected. If you're certain, you can also click on the link in the *Source string location* section and submit a PR with your correction.

3.1.2 Contribute to Weblate documentation

You are welcome to improve the documentation page of your choice. Do it easily by clicking the *Edit on GitHub* button in the top-right corner of the page.

Please respect these guidelines while writing:

1. Don't remove part of the documentation if it's valid.
2. Use clear and easily-understandable language. You are writing tech docs, not a poem. Not all docs readers are native speakers, be thoughtful.
3. Don't be afraid to ask if you are not certain. If you have to ask about some feature while editing, don't change its docs before you have the answer. This means: You change or ask. Don't do both at the same time.
4. Verify your changes by performing described actions while following the docs.
5. Send PR with changes in small chunks to make it easier and quicker to review and merge.
6. If you want to rewrite and change the structure of a big article, do it in two steps:
 1. Rewrite
 2. Once the rewrite is reviewed, polished, and merged, change the structure of the paragraphs in another PR.

Consejo: You can [translate the docs](#).

3.1.3 Extending built-in language definitions

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

You are welcome to add missing language definitions to `languages.csv`, other files are generated from that file.

3.1.4 Weblate discussions

If you have an idea and not sure if it's suitable for an issue, don't worry. You can join the community in [GitHub discussions](#).

3.1.5 Funding Weblate development

You can boost Weblate's development on the [donate page](#). Funds collected there are used to enable gratis hosting for libre software projects and further development of Weblate. Please check the [donate page](#) for options, such as funding goals and the rewards you get as a proud funder.

Supporters 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
- MyExpenses.Mobi

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 running only tests in the `weblate.machine` module:

```
./rundev.sh test --failfast weblate.machine
```

Nota: Be careful that your Docker containers are up and running before running the tests. You can check that by running the `docker ps` command.

Para mostrar los registros:

```
./rundev.sh logs
```

Para detener los contenedores en segundo plano, ejecute:

```
./rundev.sh stop
```

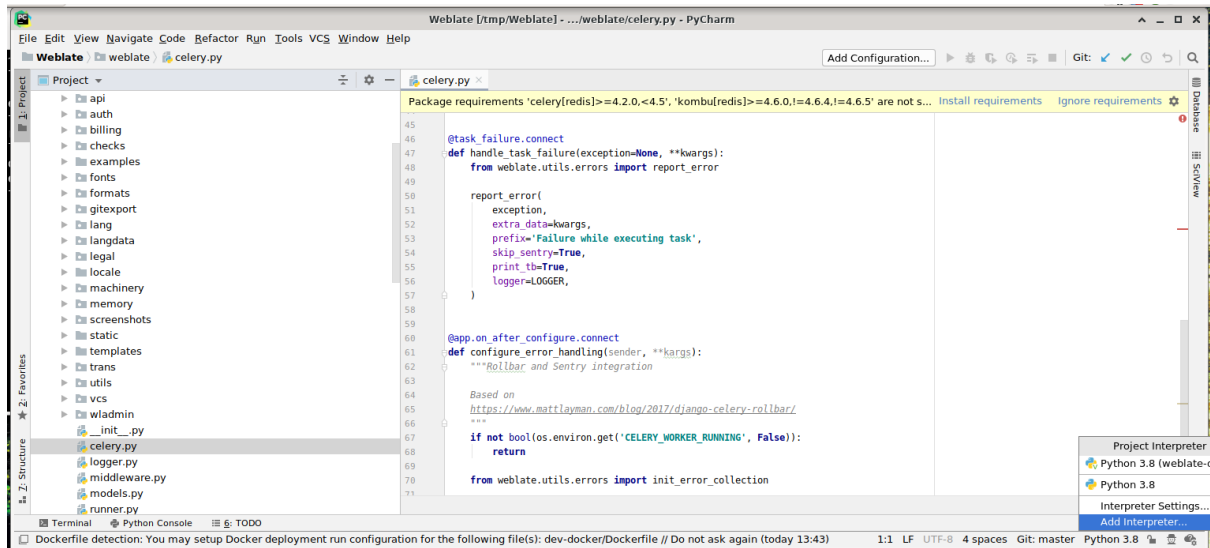
Ejecutar la secuencia de órdenes sin argumentos recreará el contenedor de Docker y lo reiniciará.

Nota: Esta no es una puesta en funcionamiento ideal para entornos de producción, puesto que incluye varios arreglos apurados inseguros que, sin embargo, facilitan el desarrollo.

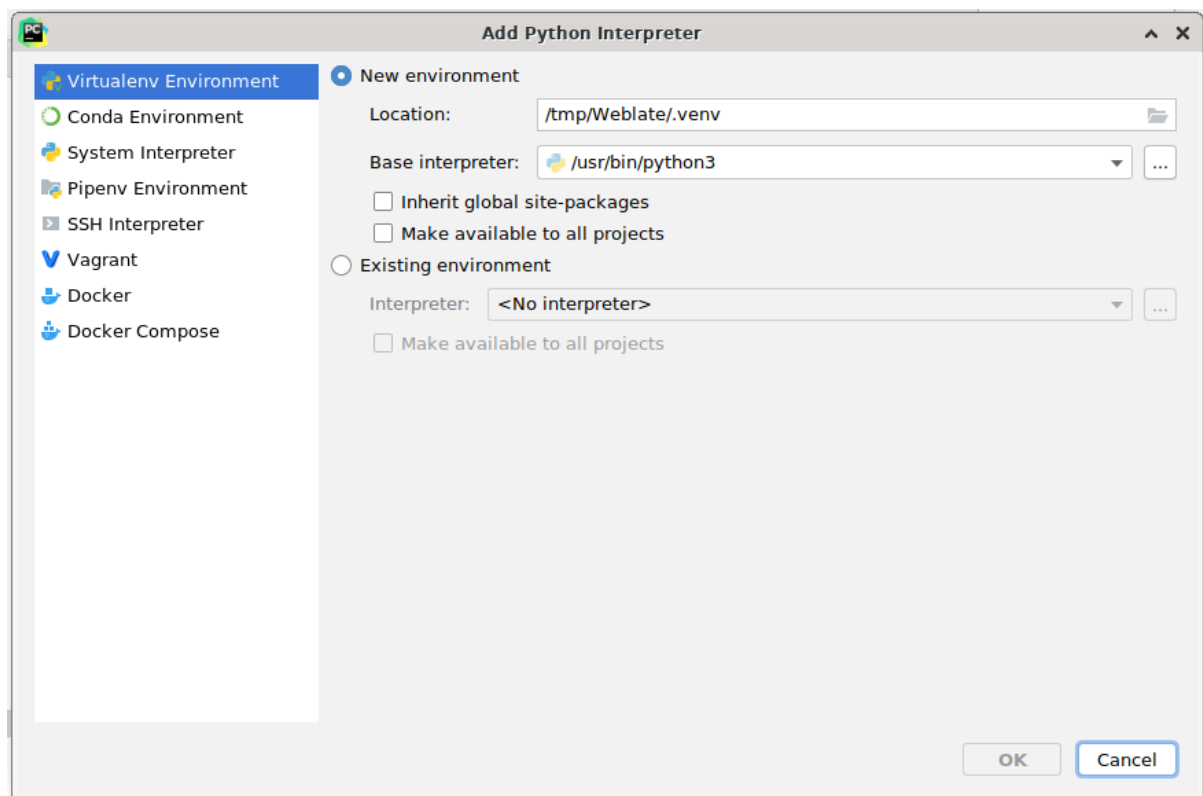
3.2.4 Trabajar en Weblate con PyCharm

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:

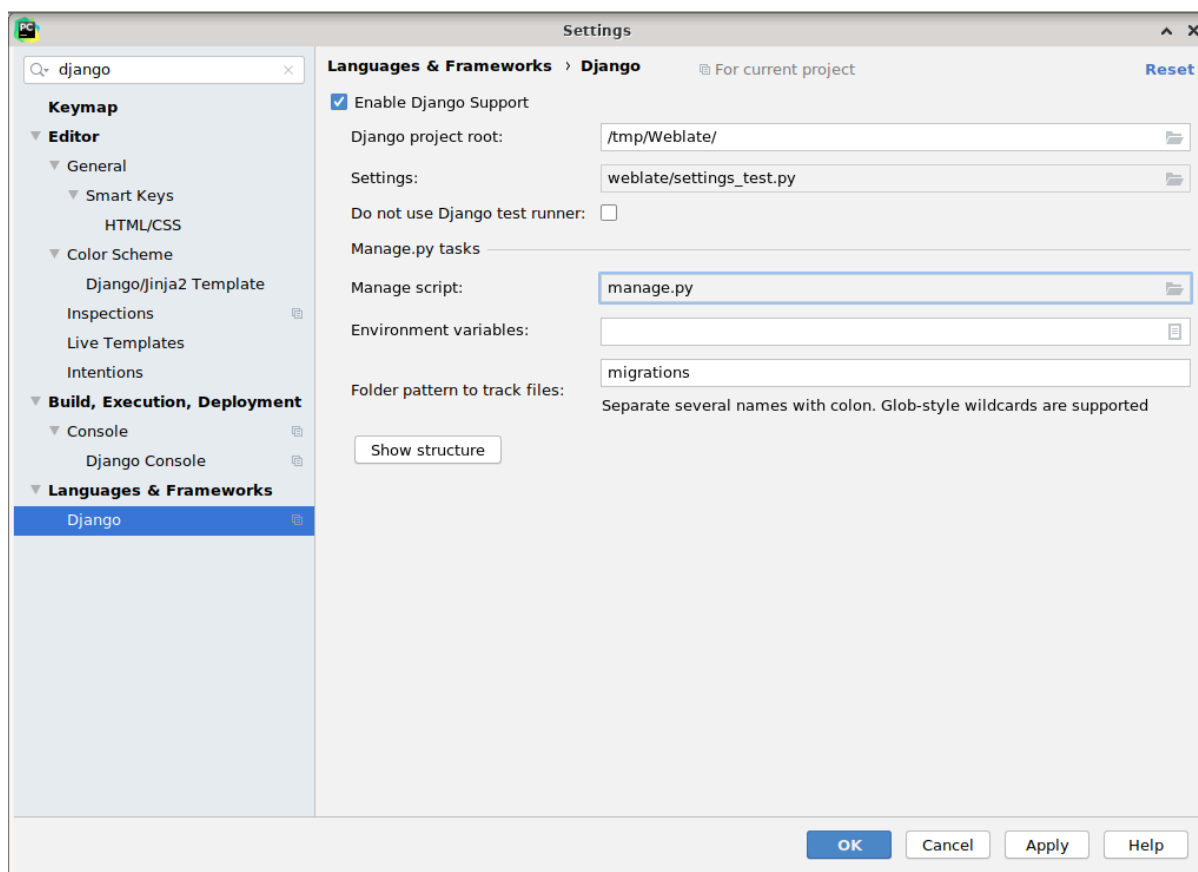


You can either choose to let PyCharm create the virtualenv for you, or select an already existing one:



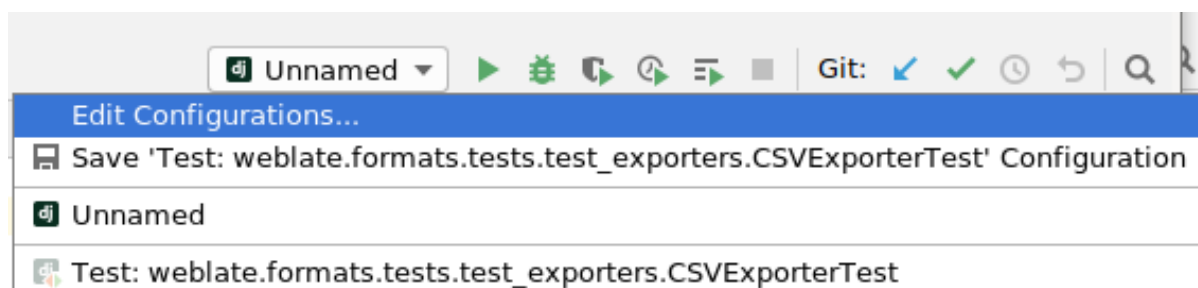
Don't forget to install the dependencies once the interpreter is set: Either through the console (the console from the IDE will directly use your virtualenv by default), or through the interface when you get a warning about missing dependencies.

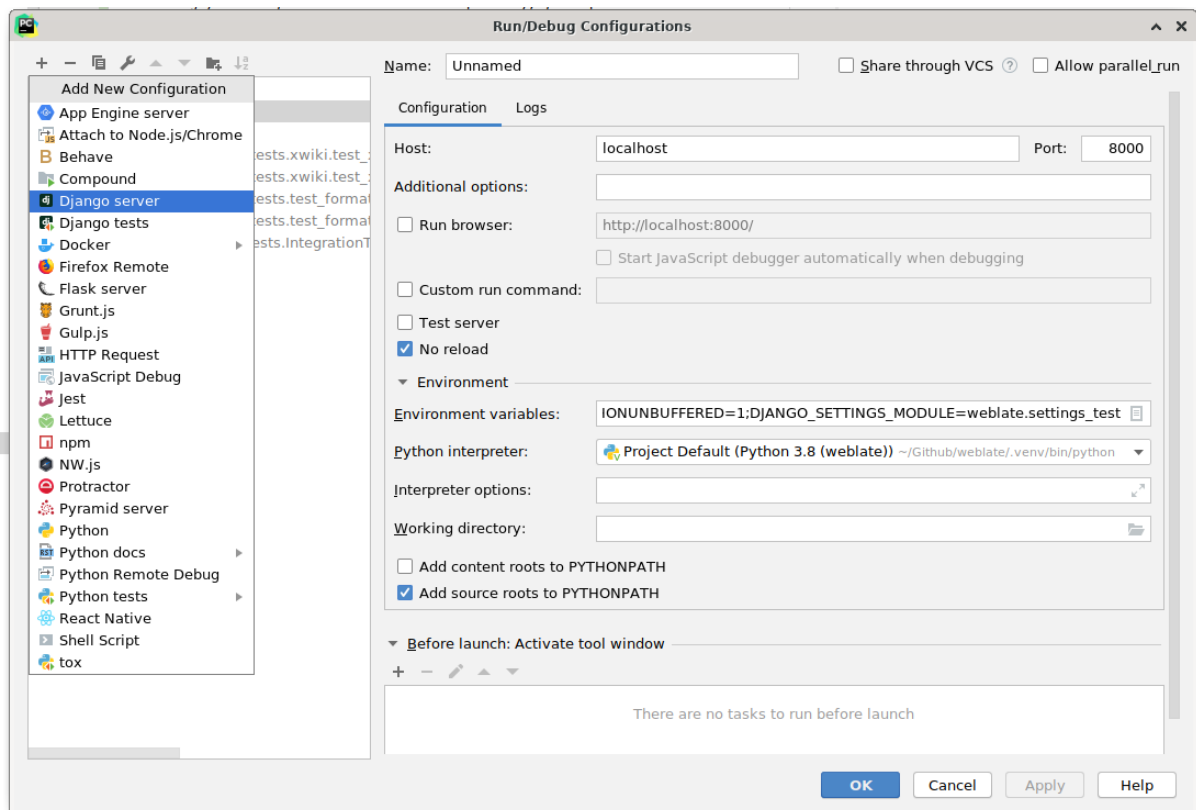
The second step is to set the right info to use Django natively inside PyCharm: The idea is to be able to immediately trigger the unit tests in the IDE. For that you need to specify the root path of the Django project and the path to its settings:



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 Coding guidelines

Cualquier código escrito para Weblate debe crearse teniendo en mente los [principios de seguridad por naturaleza](#).

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize [PEP 484](#) type hints. We're not checking this in our CI yet as existing code does not yet include them.

3.3.2 Coding standard and linting the code

The code should follow PEP-8 coding guidelines and should be formatted using **black** code formatter.

To check the code quality, you can use **flake8**, the recommended plugins are listed in `.pre-commit-config.yaml` and its configuration is placed in `setup.cfg`.

The easiest approach to enforce all this is to install [pre-commit](#). The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the `requirements-lint.txt`) turn it on by running `pre-commit install` in Weblate checkout. This way all your changes will be automatically checked.

You can also trigger check manually, to check all files run:

```
pre-commit run --all
```

3.4 Depurar Weblate

Bugs can behave as application crashes or as various misbehavior. You are welcome to collect info on any such issue and submit it to the [issue tracker](#).

3.4.1 Modo de depuración

Turning on debug mode will make the exceptions show in the web browser. This is useful to debug issues in the web interface, but not suitable for a production environment because it has performance consequences and might leak private data.

In a production environment, use [ADMINS](#) to receive e-mails containing error reports, or configure error collection using a third-party service.

Ver también:

[Desactivar el modo de depuración](#), [Properly configure admins](#), [Collecting error reports](#)

3.4.2 Registros de Weblate

Weblate can produce detailed logs of what is going on in the background. In the default configuration it uses syslog and that makes the log appear either in `/var/log/messages` or `/var/log/syslog` (depending on your syslog daemon configuration).

The Celery process (see [Tareas en segundo plano con Celery](#)) usually produces its own logs as well. The example system-wide setups logs to several files under `/var/log/celery/`.

Docker containers log to their output (as per usual in the Docker world), so you can look at the logs using `docker-compose logs`.

Ver también:

[Configuración de muestra](#) contains `LOGGING` configuration.

3.4.3 Not processing background tasks

A lot of things are done in the background by Celery workers. If things like sending out e-mails or component removal does not work, there might be a related issue.

Elementos que revisar en este caso:

- Check that the Celery process is running, see *Tareas en segundo plano con Celery*
- Check the Celery queue status, either in *Interfaz de gestión*, or using `celery_queues`
- Look in the Celery logs for errors (see *Registros de Weblate*)

3.4.4 Not receiving e-mails from Weblate

You can verify whether outgoing e-mail is working correctly by using the `sendtestemail` management command (see *Invocar órdenes de gestión* for instructions on how to invoke it in different environments) or by using *Interfaz de gestión* under the *Tools* tab.

These send e-mails directly, so this verifies that your SMTP configuration is correct (see *Configurar el correo electrónico saliente*). Most of the e-mails from Weblate are however sent in the background and there might be some issues with Celery involved as well, please see *Not processing background tasks* for debugging that.

3.4.5 Analyzing application crashes

In case the application crashes, it is useful to collect as much info about the crash as possible. This can be achieved by using third-party services which can collect such info automatically. You can find info on how to set this up in *Collecting error reports*.

3.4.6 Silent failures

Lots of tasks are offloaded to Celery for background processing. Failures are not shown in the user interface, but appear in the Celery logs. Configuring *Collecting error reports* helps you to notice such failures easier.

3.4.7 Problemas de rendimiento

In case Weblate performs badly in some scenario, please collect the relevant logs showing the issue, and anything that might help figuring out where the code might be improved.

In case some requests take too long without any indication, you might want to install `dogslow` along with *Collecting error reports* and get pinpointed and detailed tracebacks in the error collection tool.

In case the slow performance is linked to the database, you can also enable logging of all database queries using following configuration after enabling `DEBUG`:

```
LOGGING["loggers"]["django.db.backends"] = {"handlers": ["console"], "level":
↪ "DEBUG" }
```

3.5 Weblate por dentro

Nota: This chapter will give you basic overview of Weblate internals.

Weblate derives most of its code structure from, and is based on [Django](#).

3.5.1 Estructura de directorios

Quick overview of directory structure of Weblate main repository:

docs Source code for this documentation, which can be built using [Sphinx](#).

dev-docker Docker code to run development server, see [Ejecutar Weblate localmente en Docker](#).

weblate Source code of Weblate as a [Django](#) application, see [Weblate por dentro](#).

weblate/static Client files (CSS, Javascript and images), see [Interfaz de Weblate](#).

3.5.2 Módulos

Weblate se compone de varias aplicaciones de Django (algunas son opcionales; vea [Módulos opcionales de Weblate](#)):

`accounts`

Cuenta de usuario, perfiles y notificaciones.

`addons`

Add-ons to tweak Weblate behavior, see [Complementos](#).

`api`

API based on [Django REST framework](#).

`auth`

Authentication and permissions.

`billing`

The optional [Facturación](#) module.

`checks`

Translation string [Comprobaciones de calidad](#) module.

`fonts`

Font rendering checks module.

`formats`

File format abstraction layer based on [translate-toolkit](#).

`gitexport`

The optional [Git exporter](#) module.

`lang`

Módulo que define los idiomas y los modelos de pluralización.

`legal`

El módulo facultativo [Información legal](#).

`machinery`

Integración de servicios de traducción automática.

memory

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 Developing add-ons

Complementos are way to customize localization workflow in Weblate.

class `weblate.addons.base.BaseAddon` (*storage=None*)

Base class for Weblate add-ons.

classmethod `can_install` (*component, user*)

Check whether add-on is compatible with given component.

configure (*settings*)

Save configuration.

daily (*component*)

Actuador desencadenado diariamente.

classmethod `get_add_form` (*user, component, **kwargs*)

Return configuration form for adding new add-on.

get_settings_form (*user, **kwargs*)

Return configuration form for this add-on.

post_add (*translation*)

Actuador desencadenado tras añadirse una traducción nueva.

post_commit (*component*)

Actuador desencadenado tras consignarse cambios en el repositorio.

post_push (*component*)

Actuador desencadenado tras enviar el repositorio a la ubicación del proyecto originario.

post_update (*component, previous_head: str, skip_push: bool*)

Actuador desencadenado tras actualizarse el repositorio desde el origen.

Parámetros

- **previous_head** (*str*) – «HEAD» del repositorio antes de la actualización; puede estar vacía durante la clonación inicial.

- **skip_push** (*bool*) – Whether the add-on operation should skip pushing changes upstream. Usually you can pass this to underlying methods as `commit_and_push` or `commit_pending`.

pre_commit (*translation, author*)

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 add-on state information.

store_post_load (*translation, store*)

Actuador desencadenado tras procesar un archivo.

It receives an instance of a file format class as a argument.

This is useful to modify file format class parameters, for example adjust how the file will be saved.

unit_pre_create (*unit*)

Actuador desencadenado antes de crearse una unidad nueva.

Here is an example add-on:

```
#
# Copyright © 2012-2022 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 add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
```

(continué en la próxima página)

(proviene de la página anterior)

```
# Detailed add-on description
description = _("This add-on does nothing it is just an example.")

# Callback to implement custom behavior
def pre_commit(self, translation, author):
    return
```

3.7 Interfaz de Weblate

La interfaz está elaborada con Bootstrap, jQuery y algunas bibliotecas de terceros.

3.7.1 Navegadores admitidos

Weblate es compatible con las versiones estables más recientes de todos los navegadores web principales, en todas las plataformas.

No se brinda servicio técnico explícito al emplear navegadores alternativos que hacen uso (directo o a través de la API de visualización web de la plataforma) de la versión más reciente de WebKit, Blink o Gecko. Sin embargo, Weblate se visualizará y funcionará correctamente en esos navegadores, en la mayoría de los casos.

Es posible que navegadores más antiguos funcionen, pero determinadas funciones se verán limitadas.

3.7.2 Gestión de dependencias

The yarn package manager is used to update third party libraries. The configuration lives in `scripts/yarn` and there is a wrapper script `scripts/yarn-update` to upgrade the libraries, build them and copy to correct locations in `weblate/static/vendor`, where all third partly frontend code is located. The Weblate specific code should be placed directly in `weblate/static` or feature specific subdirectories (for example `weblate/static/editor`).

Adding new third-party library typically consists of:

```
# Add a yarn package
yarn --cwd scripts/yarn add PACKAGE
# Edit the script to copy package to the static folder
edit scripts/yarn-update
# Run the update script
./scripts/yarn-update
# Add files to git
git add .
```

3.7.3 Coding style

Weblate relies on [Prettier](#) for the code formatting for both JavaScript and CSS files.

We also use [ESLint](#) to check the JavaScript code.

3.7.4 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

El seguimiento de errores de Weblate se aloja en GitHub.

Feel welcome to report any issues you have, or suggest improvement for Weblate there. There are various templates prepared to comfortably guide you through the issue report.

If what you have found is a security issue in Weblate, please consult the [Problemas de seguridad](#) section below.

If you are not sure about your bug report or feature request, you can try [Weblate discussions](#).

3.8.1 Problemas de seguridad

In order to give the community time to respond and upgrade, you are strongly urged to report all security issues privately. HackerOne is used to handle security issues, and can be reported directly at [HackerOne](#). Once you submit it there, community has limited but enough time to solve the incident.

Alternatively, report to security@weblate.org, which ends up on HackerOne as well.

If you don't want to use HackerOne, for whatever reason, you can send the report by e-mail to michal@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
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

Prior to running tests you should collect static files as some tests rely on them being present:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

También puede especificar que se ejecuten solo pruebas individuales:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Consejo: The tests can also be executed inside developer docker container, see *Ejecutar Weblate localmente en Docker*.

Ver también:

See *Testing in Django* for more info on running and writing tests for Django.

3.10 Esquemas de datos

Weblate uses *JSON Schema* to define layout of external JSON files.

3.10.1 Weblate Translation Memory Schema

https://weblate.org/schemas/weblate-memory.schema.json			
type	array		
items	The Translation Memory Item		
	type	object	
	properties		
	• category	The String Category	
		1 is global, 2 is shared, 10000000+ are project specific, 20000000+ are user specific	
		type	integer
		examples	1
		minimum	0
		default	1
		• origin	The String Origin
	Filename or component name		
	type		string
	examples		test.tmx
			proyecto/componente
default			

continué en la próxima página

Tabla 1 – proviene de la página anterior

definiciones	• source	<i>The Source String</i>	
		type	<i>string</i>
		examples	Hello
		minLength	1
		default	
	• source_language	<i>The Source Language</i>	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		type	<i>string</i>
		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

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	Basic		
	type	object	
	properties		
	• username	Nombre de usuario	
		type	string
		examples	administrador
		default	
	• full_name	Nombre completo	
		type	string
		examples	Weblate Admin
		default	
	• email	Correo electrónico	
		type	string
		examples	noreply@example.com
		default	
	• date_joined	Fecha de incorporación	
		type	string
		examples	2019-11-18T18:53:54.862Z
		default	
• profile	Perfil		
	type	object	

continué en la próxima página

Tabla 2 – proviene de la página anterior

	properties		
	• language	<i>Idioma</i>	
		type	<i>string</i>
		examples	cs
		pauta	^.*\$
		default	
	• suggested	<i>Number of suggested strings</i>	
		type	<i>integer</i>
		examples	1
		default	0
	• translated	<i>Number of translated strings</i>	
		type	<i>integer</i>
		examples	24
		default	0
	• uploaded	<i>Number of uploaded screenshots</i>	
		type	<i>integer</i>
		examples	1
		default	0
	• hi-de-completed	<i>Hide completed translations on the dashboard</i>	
		type	<i>boolean</i>
		examples	Falso
		default	Verdadero
	• secon-dary_in_zen	<i>Mostrar traducciones secundarias en modo zen</i>	
		type	<i>boolean</i>
		examples	Verdadero
		default	Verdadero
	• hi-de_source_secondary	<i>Ocultar origen si existe traducción secundaria</i>	
		type	<i>boolean</i>
		examples	Falso
		default	Verdadero
	• editor_link	<i>Enlace del editor</i>	
		type	<i>string</i>
		examples	
		pauta	^.*\$
		default	
	• transla-te_mode	<i>Modo del editor de traducciones</i>	
		type	<i>integer</i>
		examples	0
		default	0
	• zen_mode	<i>Modo de editor zen</i>	
		type	<i>integer</i>
		examples	0
		default	0
	• spe-cial_chars	<i>Caracteres especiales</i>	
		type	<i>string</i>
		examples	
		pauta	^.*\$
		default	
	• dash-board_view	<i>Vista predeterminada del cuadro de mando</i>	
		type	<i>integer</i>
		examples	1
		default	0
	• dash-board_components	<i>Lista de componentes predeterminados</i>	
		default	null
		anyOf	type
		type	<i>null</i>
		type	<i>integer</i>

continué en la próxima página

Tabla 2 – proviene de la página anterior

	• languages	<i>Idiomas traducidos</i>			
		type	array		
		default			
		items	<i>Código de idioma</i>		
			type	string	
			examples	cs	
			pauta	^.*\$	
			default		
		• secondary_languages	<i>Idiomas secundarios</i>		
			type	array	
	default				
	items		<i>Código de idioma</i>		
			type	string	
			examples	sk	
			pauta	^.*\$	
			default		
	• monitorizado	<i>Proyectos monitorizados</i>			
		type	array		
		default			
		items	«Slug» del proyecto		
			type	string	
examples			weblate		
pauta			^.*\$		
default					
• Registro de auditoría	<i>Registro de auditoría</i>				
	type	array			
	default				
	items	<i>Elementos</i>			
		type	object		
		properties			
		• Dirección	<i>Dirección IP</i>		
			type	string	
			examples	127.0.0.1	
			pauta	^.*\$	
			default		
		• Agente_de usuario	<i>Agente de usuario</i>		
			type	string	
			examples	PC / Linux / Firefox 70.0	
			pauta	^.*\$	
			default		
		• cronomarcador	<i>Cronomarcador</i>		
			type	string	
			examples	2019-11-18T18:58:30.845Z	
			pauta	^.*\$	
			default		
• actividad		<i>Actividad</i>			
		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 -j 12 -C docs update-screenshots`.
4. Merge any possibly pending translations `wlc push; git remote update; git merge origin/weblate`

Perform the release:

5. Create a release `./scripts/create-release --tag` (see below for requirements).

Post release manual steps:

6. Update Docker image.
7. Close GitHub milestone.
8. Once the Docker image is tested, add a tag and push it.
9. Update Helm chart to new version.
10. Include new version in `.github/workflows/migrations.yml` to cover it in migration testing.
11. Increase version in the website download links.
12. Increase version in the repository by `./scripts/set-version`.

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

The Docker containers are regularly scanned using [Anchore](#) and [Trivy](#) security scanners.

This allows us to detect vulnerabilities early and release improvements quickly.

You can get the results of these scans at GitHub — they are stored as artifacts on our CI in the SARIF format (Static Analysis Results Interchange Format).

Ver también:

Integración continua

3.13 Contributing to Weblate modules

Besides the main repository, Weblate consists of several Python modules. All these follow same structure and this documentation covers them all.

For example, this covers:

- [wlc](#), Python client library, see *Cliente de Weblate*
- [translation-finder](#), used to discover translatable files in the repository
- [language-data](#), language definitions for Weblate, see *Definiciones de idioma*

3.13.1 Coding guidelines

Cualquier código escrito para Weblate debe crearse teniendo en mente los [principios de seguridad por naturaleza](#).

Any code should come with documentation explaining the behavior. Don't forget documenting methods, complex code blocks, or user visible features.

Any new code should utilize [PEP 484](#) type hints. We're not checking this in our CI yet as existing code does not yet include them.

3.13.2 Running tests

The tests are executed using `py.test`. First you need to install test requirements:

```
pip install -r requirements-test.txt
```

You can then execute the testsuite in the repository checkout:

```
py.test
```

Ver también:

The CI integration is very similar to *Weblate testsuite and continuous integration*.

3.13.3 Coding standard and linting the code

The code should follow PEP-8 coding guidelines and should be formatted using **black** code formatter.

To check the code quality, you can use **flake8**, the recommended plugins are listed in `.pre-commit-config.yaml` and its configuration is placed in `setup.cfg`.

The easiest approach to enforce all this is to install `pre-commit`. The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the `requirements-lint.txt`) turn it on by running `pre-commit install` in Weblate checkout. This way all your changes will be automatically checked.

You can also trigger check manually, to check all files run:

```
pre-commit run --all
```

Ver también:

Código fuente de Weblate

3.14 Acerca de Weblate

3.14.1 Objetivos del proyecto

Web-based continuous localization tool with tight *Integración de control de versiones* supporting a wide range of *file formats*, making it easy for translators to contribute.

3.14.2 Nombre del proyecto

«Weblate» es una palabra híbrida creada a partir de las voces inglesas «web» y «translate».

3.14.3 Sitio web del proyecto

The landing page is <https://weblate.org> and there is a cloud-hosted service at <https://hosted.weblate.org>. The documentation can be read at <https://docs.weblate.org>.

3.14.4 Logotipos del proyecto

The project logos and other graphics are available in <https://github.com/WeblateOrg/graphics>.

3.14.5 Leadership

This project is maintained by Michal Čihař, who can be reached at michal@cihar.com.

3.14.6 Autores

Weblate was started by Michal Čihař. Since its inception in 2012, thousands of people have contributed.

3.15 Licencia

Copyright © 2012–2022 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/>](https://www.gnu.org/licenses/).

4.1 Weblate 4.12.1

Released on April 29th 2022.

- Fixed pull request message title.
- Improved syntax error handling in Fluent format.
- Fixed avatar display in notification e-mails.
- Add support for web monetization.
- Fixed removal of stale source strings when removing translations.

[All changes in detail.](#)

4.2 Weblate 4.12

Released on April 20th 2022.

- Added support for Amharic in *Punto final desigual*.
- Added support for Burmese in *Signo de interrogación desigual*.
- Extended options of the *Generación de pseudorregiones* add-on.
- Added `ignore-all-checks` flag to ignore all quality checks on a string.
- Avoid *Generación de pseudorregiones* add-on to trigger failing checks.
- Added support for *Gitea pull requests*.
- Added Linux style language code to *Estilo de código de idioma*.
- Added support for rebuilding project translation memory.
- Improved API for creating components from a file.
- Add copy and clone buttons to other translations.
- Make merge request message configurable at component level.
- Improved maximal length restriction behavior with XML tags.

- Fixed loading Fluent files with additional comments.

[All changes in detail.](#)

4.3 Weblate 4.11.2

Released on March 4th 2022.

- Fixed corrupted MO files in the binary release.

[All changes in detail.](#)

4.4 Weblate 4.11.1

Released on March 4th 2022.

- Fixed missing sanitizing of arguments to Git and Mercurial - CVE-2022-23915, see [GHSA-3872-f48p-pxqj](#) for more details.
- Fixed loading fuzzy strings from CSV files.
- Added support for creating teams using the API.
- Fixed user mention suggestions display.
- The project tokens access can now be customized.

[All changes in detail.](#)

4.5 Weblate 4.11

Released on February 25th 2022.

- Fixes stored XSS - CVE-2022-24710, see [GHSA-6jp6-9rf9-gc66](#) for more details.
- Se arregló la instalación de complementos a través de la API.
- Se cambió el nombre de *Cadenas que necesitan intervención* a *Cadenas sin terminar*.
- Se corrigieron falsos positivos de *Sintaxis MessageFormat de ICU*.
- Indicate lock and contributor agreement on other occurrences listing.
- Fixed updating PO files with obsolete strings or missing plurals.
- Improved squash add-on compatibility with Gerrit.
- Automatically initialize user languages based on the [Accept-Language](#) header.
- Improved error handling on string removal.
- Weblate now requires Python 3.7 or newer.
- Fixed some write operations with project token authentication.
- Fixed string state tracking when the strings changes in the repository.
- Track string changes from the repository.
- Sticky header on translations listing to improve navigation.
- Fixed untranslating strings in *Propiedades de Java*.
- Fixed Git operation with non-ascii branch names.
- New add-on *Prerrellenar traducción con original*.

- Added *Merge without fast-forward* *Estilo de fusión*.
- Fixed *Traducción automática* add-on trigger on newly added strings.
- Improved punctuation checks for Burmese.
- Added support for defining custom teams at project level to grant users access, see *Gestionar el control de acceso por proyecto*.
- Added documentation links to alerts.
- Docker container automatically enables TLS/SSL for outgoing e-mail when needed.
- Added support for searching for resolved comments.
- Added support for borgbackup 1.2.
- Fixed applying of *Automatically translated* label.

[All changes in detail.](#)

4.6 Weblate 4.10.1

Publicada el 22 de diciembre de 2021.

- Se documentaron los cambios introducidos por la actualización a Django 4.0.
- Fixed displaying of *Automatically translated* label.
- Fixed API display of branch in components with a shared repository.
- Improved analysis on the failed push alert.
- Fixed manually editing page when browsing changes.
- Se mejoró la precisión de *Kashida utilizado*.
- El contenedor Docker de Weblate ahora utiliza Python 3.10.

[All changes in detail.](#)

4.7 Weblate 4.10

Publicada el 16 de diciembre de 2021.

- Se añadió compatibilidad con niveles de formalidad y sustitutorios con DeepL.
- Bulk edit and search and replace are now available on project and language level.
- Se añadió la posibilidad de filtrar en la función Buscar y reemplazar.
- Fixed: «Perform automatic translation» privilege is no longer part of the *Languages* group.
- «Perform automatic translation» is in the *Administration* and the new *Automatic translation* group.
- Fixed generating XLSX files with special chars.
- Added ability to the GitHub authentication backend to check if the user belongs to a specific GitHub organization or team.
- Improved feedback on invalid parameters passed to API.
- Added support for project scoped access tokens for API.
- Fixed string removal in some cases.
- Fixed translating newly added strings.
- Label automatically translated strings to ease their filtering.

[All changes in detail.](#)

4.8 Weblate 4.9.1

Publicada el 19 de noviembre de 2021.

- Fixed upload of monolingual files after changing template.
- Improved handling of whitespace in flags.
- Add support for filtering in download API.
- Fixed statistics display when adding new translations.
- Mitigate issues with GitHub SSH key change.

[All changes in detail.](#)

4.9 Weblate 4.9

Publicada el 10 de noviembre de 2021.

- Provide more details for events in history.
- Improved rendering of history.
- Improved performance of the translation pages.
- Added support for restricting translation file downloads.
- The `safe-html` can now understand Markdown when used with `md-text`.
- The `max-length` tag now ignores XML markup when used with `xml-text`.
- Fixed dimensions of rendered texts in *Tamaño máximo de la traducción*.
- Lowered app store title length to 30 to assist with upcoming Google policy changes.
- Added support for customizing SSH invocation via `SSH_EXTRA_ARGS`.
- Se añadieron comprobaciones para el MessageFormat de ICU.
- Improved error condition handling in machine translation backends.
- Highlight unusual whitespace characters in the strings.
- Added option to stay on translated string while editing.
- Added support for customizing Borg invocation via `BORG_EXTRA_ARGS`.
- Fixed generating of MO files for monolingual translations.
- Added API endpoint to download all component translations as a ZIP file.
- Added support for Python 3.10.
- Added support for resending e-mail invitation from the management interface.

[All changes in detail.](#)

4.10 Weblate 4.8.1

Released on September 10th 2021.

- Fixed user removal in Django admin interface.
- Document add-on parameters in greater detail.
- Fixed JavaScript error in glossary.
- Add limit to number of matches in consistency check.
- Improve handling of placeholders in machine translations.
- Fixed creating add-ons using API.
- Added `PRIVACY_URL` setting to add privacy policy link to the footer.
- Hide member e-mail addresses from project admins.
- Improved gettext PO merging in case of conflicts.
- Improved glossary highlighting.
- Improved `safe-html` flag behavior with XML checks.
- Fixed commit messages for linked components.

[All changes in detail.](#)

4.11 Weblate 4.8

Released on August 21th 2021.

- Added support for Apple stringsdict format.
- The exact search operator is now case-sensitive with PostgreSQL.
- Fixed saving glossary explanations in some cases.
- Documentation improvements.
- Performance improvements.
- Improved squash add-on compatibility with Gerrit.
- Fixed adding strings to monolingual glossary components.
- Improved performance in handling variants.
- Fixed squash add-on sometimes skipping parsing upstream changes.
- Preserve file extension for downloads.
- Added support for the Fluent format.
- Added support for using tabs to indent JSON formats.

[All changes in detail.](#)

4.12 Weblate 4.7.2

Released on July 15th 2021.

- Support more language aliases to be configured on a project.
- Fixed search string validation in API.
- Fixed Git exporter URLs after a domain change.
- Fixed cleanup add-on for Windows RC files.
- Fixed possible crash in XLIFF updating.

[All changes in detail.](#)

4.13 Weblate 4.7.1

Released on June 30th 2021.

- Improved popup for adding terms to glossary.
- Added support for LibreTranslate machine translation service.
- Added rate limiting on creating new projects.
- Improved performance of file updates.

[All changes in detail.](#)

4.14 Weblate 4.7

Released on June 17th 2021.

- Improved configuration health check.
- Added support for `object-pascal-format` used in gettext PO, see *Formato de Object Pascal*.
- Cambio de nombre de *Claves cercanas* a *Claves similares* para describir mejor el propósito.
- Se añadió compatibilidad con *mi18n lang files*.
- Se mejoró la integración con la autenticación SAML.
- Fixed *Gerrit* integration to better handle corner cases.
- Weblate now requires Django 3.2.
- Fixed inviting users when e-mail authentication is disabled.
- Improved language definitions.
- Added support for blocking users from contributing to a project.
- Fixed automatic creation of glossary languages.
- Extended documentation about add-ons.
- Performance improvements for components with linked repositories.
- Added support for free DeepL API.
- The user management no longer needs Django admin interface.

[All changes in detail.](#)

4.15 Weblate 4.6.2

Released on May 8th 2021.

- Fixed crash after moving shared component between projects.
- Fixed adding new strings to empty properties files.
- Fixed copy icon alignment in RTL languages.
- Extended string statistics on the Info tab.
- Fixed handling of translation files ignored in Git.
- Improved metrics performance.
- Fixed possible bug in saving glossaries.
- Fixed consistency check behavior on languages with different plural rules.

[All changes in detail.](#)

4.16 Weblate 4.6.1

Released on May 2nd 2021.

- Remove obsolete spam protection code.
- Improve source plural check accuracy.
- Update list of user interface languages in Docker.
- Improved error messages when creating pull requests.
- Fixed creating pull requests on Pagure.
- Fixed triggering automatically installed add-ons.
- Fixed possible caching issues on upgrade.
- Fixed adding new units to monolingual translations using upload.

[All changes in detail.](#)

4.17 Weblate 4.6

Released on April 19th 2021.

- The `auto_translate` management command has now a parameter for specifying translation mode.
- Added support for *Archivos de texto*.
- Added trends and metrics for all objects.
- Added support for directly copying text from secondary languages.
- Added date filtering when browsing changes.
- Se remozaron los gráficos de actividad.
- Sender for contact form e-mails can now be configured.
- Improved parameters validation in component creation API.
- The rate limiting no longer applies to superusers.
- Improved automatic translation add-on performance and reliability.

- The rate limiting now can be customized in the Docker container.
- API for creating components now automatically uses *URL internos de Weblate*.
- Indicación de estado simplificada en las listas de cadenas.
- Password hashing now uses Argon2 by default.
- Barras de progreso simplificadas que indican el estado de la traducción.
- Renamed *Adición de idiomas faltantes* to clarify the purpose.
- Fixed saving string state to XLIFF.
- Added language-wide search.
- Initial support for *Scaling horizontally* the Docker deployment.

[All changes in detail.](#)

4.18 Weblate 4.5.3

Publicada el 1.º de abril de 2021.

- Fixed metrics collection.
- Fixed possible crash when adding strings.
- Improved search query examples.
- Fixed possible loss of newly added strings on replace upload.

4.19 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.](#)

4.20 Weblate 4.5.1

Released on March 5th 2021.

- Fixed editing of glossary flags in some corner cases.
- Extend metrics usage to improve performance of several pages.
- Se almacena el idioma de origen correcto en los archivos TMX.
- Better handling for uploads of monolingual PO using API.
- Improved alerts behavior on glossary components.
- Improved Markdown link checks.
- Indicate glossary and source language in breadcrumbs.
- Paginated component listing of huge projects.
- Improved performance of translation, component or project removal.
- Improved bulk edit performance.
- Fixed preserving «Needs editing» and «Approved» states for ODF files.
- Improved interface for customizing translation-file downloads

[All changes in detail.](#)

4.21 Weblate 4.5

Publicada el 19 de febrero de 2021.

- Se añadió compatibilidad con `lua-format` tal como se usa en los archivos PO de gettext.
- Se agregó la posibilidad de compartir un componente entre proyectos.
- Se arregló el comportamiento de la comprobación de varias variables sin nombre en diversos indicadores de formato.
- Se desechó el campo Lista de correo en los proyectos en favor de instrucciones para traductores genéricas.
- Added pseudolocale generation add-on.
- Se pueden utilizar ahora archivos TermBase eXchange.
- Se añadió la posibilidad de definir manualmente las variantes de una cadena a través de un indicador.
- Se mejoró el desempeño de las comprobaciones de coherencia.
- Se mejoró el desempeño de la memoria de traducción en cadenas extensas.
- Se permite efectuar búsquedas en las explicaciones.
- Ahora se pueden añadir y quitar cadenas también en los formatos bilingües.
- Se amplió la lista de idiomas admitidos en la traducción automática de Amazon Translate.
- Activar automáticamente las comprobaciones de MessageFormat de Java para las `.properties` de Java.
- Se añadió un método de carga nuevo para añadir cadenas nuevas a una traducción.
- Se añadió una interfaz simple para explorar las traducciones.
- Los glosarios ahora se almacenan como componentes normales.
- Se eliminó la API específica de los glosarios, dado que ahora se utiliza la API de los componentes para el mismo fin.
- Se añadió una interfaz simplificada para activar o desactivar determinados indicadores.

- Se permite marcar términos como de no traducción o como prohibidos en los glosarios.
- Se añadió la capacidad de definir terminología en un glosario.
- Se movió el control de dirección del texto para obtener más espacio para el teclado virtual.
- Se añadió una opción para monitorizar automáticamente los proyectos en los que se contribuye.
- Se añadió una comprobación para asegurar que las traducciones coincidan con el glosario.
- Se permite personalizar el color de texto de la navegación.

[All changes in detail.](#)

4.22 Weblate 4.4.2

Publicada el 14 de enero de 2021.

- Se arregló el daño en un archivo MO distribuido.

4.23 Weblate 4.4.1

Publicada el 13 de enero de 2021.

- Se corrigió la reversión de cambios con plurales.
- Se arregló la visualización de la ayuda en la configuración del proyecto.
- Se perfeccionó la administración de usuarios.
- Se manipulan mejor los contextos de los archivos PO monolingües.
- Fixed cleanup add-on behavior with HTML, ODF, IDML and Windows RC formats.
- Se arregló el procesamiento de la ubicación en los archivos CSV.
- Se utiliza compresión de contenido en las descargas de archivos.
- Se mejoró la experiencia de usuario al importar a partir de un archivo ZIP.
- Se mejoró la detección de formato de archivo durante las cargas.
- Se evita la duplicación de las solicitudes de incorporación en Pagine.
- Se mejoró el rendimiento al mostrar traducciones fantasma.
- Se reemplazó el editor de traducciones para que utilice el elemento «textarea» nativo del navegador.
- Fixed cleanup add-on breaking adding new strings.
- Added API for add-ons.

[All changes in detail.](#)

4.24 Weblate 4.4

Publicada el 15 de diciembre de 2020.

- Se perfeccionó la validación al crear componentes.
- Weblate ahora requiere Django 3.1.
- Ahora es posible personalizar la apariencia en la interfaz de gestión.
- Fixed read-only state handling in bulk edit.

- Se mejoró la integración con CodeMirror.
- Added add-on to remove blank strings from translation files.
- Ahora se utiliza el editor CodeMirror para las traducciones.
- Coloración sintáctica en el editor de traducciones para XML, HTML, Markdown y reStructuredText.
- 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.](#)

4.25 Weblate 4.3.2

Publicada el 4 de noviembre de 2020.

- Fixed crash on certain component file masks.
- Improved accuracy of the consecutive duplicated words check.
- Added support for Pagure pull requests.
- Se mejoraron los mensajes de error al producirse errores en el alta.
- Se revirtió la representación de los comentarios de los programadores como Markdown.
- Se simplificó la puesta en marcha de los repositorios Git con ramas predeterminadas distintas de «master».
- Los repositorios internos de nueva creación ahora utilizan «main» como rama predeterminada.
- Reduced false positives rate of unchanged translation while translating reStructuredText.
- Se arreglaron problemas de visualización de CodeMirror en algunas situaciones.
- Se cambió el nombre del grupo Plantilla a «Orígenes» para clarificar su significado.
- Fixed GitLab pull requests on repositories with longer paths.

[All changes in detail.](#)

4.26 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.](#)

4.27 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.](#)

4.28 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.

4.29 Weblate 4.2.1

Publicada el 21 de agosto de 2020.

- Se corrigió el guardado de plurales en recursos de Android en algunas configuraciones regionales.
- Fixed crash in the cleanup add-on for some XLIFF files.
- Se permite configurar la CDN en la imagen de Docker.

4.30 Weblate 4.2

Publicada el 18 de agosto de 2020.

- Páginas de usuario mejoradas y adición de listados de usuarios.
- Se eliminó la posibilidad de migrar desde las versiones 3.x; puede migrar a través de la 4.1 o la 4.0.
- Se añadieron exportaciones a varios formatos monolingües.
- Se remozaron los gráficos de actividad.
- Es posible configurar el número de cadenas cercanas mostradas.
- Se permite bloquear los componentes cuando se producen errores en el repositorio.
- Se simplificó la navegación principal (se sustituyeron los botones por iconos).
- Se mejoró el tratamiento de los códigos de idioma en la integración con Google Translate.
- The Git squash add-on can generate `Co-authored-by`: trailers.
- Ha mejorado el procesador de consultas de búsqueda.
- Improved user feedback from format strings checks.
- Improved performance of bulk state changes.

- Added compatibility redirects after project or component renaming.
- Se añadieron notificaciones sobre aprobaciones de cadenas, bloqueos de componentes y cambios de licencia.
- Se añadió compatibilidad con ModernMT.
- Allow to avoid overwriting approved translations on file upload.
- Dropped support for some compatibility URL redirects.
- Se añadió una comprobación para los literales de plantillas ECMAScript.
- Se añadió la posibilidad de monitorizar componentes.
- Removed leading dot from JSON unit keys.
- Se eliminó la cola de Celery dedicada a la memoria de traducción.
- Se permite traducir todos los componentes en un idioma a la vez.
- Se permite configurar cabeceras `Content-Security-Policy` de HTTP.
- Added support for aliasing languages at project level.
- New add-on to help with HTML or JavaScript localization, see *[CDN de regionalización de JavaScript](#)*.
- El dominio de Weblate se establece ahora en la configuración; vea *[SITE_DOMAIN](#)*.
- Se permite buscar por componente y proyecto.

4.31 Weblate 4.1.1

Publicada el 19 de junio de 2020.

- Fixed changing autofix or add-ons configuration in Docker.
- Fixed possible crash in «About» page.
- Improved installation of byte-compiled locale files.
- Se arregló la adición de palabras al glosario.
- Se corrigieron los atajos de teclado de la automatización.
- Se quitó la salida de depuración que causaba que se descartasen sucesos del registro en algunas configuraciones.
- La vista de bloqueo en la enumeración de los proyectos ahora funciona como es debido.
- Se arregló la visualización de las claves GPG en determinados montajes.
- Se permite seleccionar qué versión de la API de DeepL utilizar.
- Added support for acting as SAML Service Provider, see *[Autenticación por SAML](#)*.

4.32 Weblate 4.1

Publicada el 15 de junio de 2020.

- Pueden crearse traducciones nuevas con código de país.
- Es posible buscar cadenas de origen con captura de pantalla.
- Información ampliada disponible en el seguimiento de estadísticas.
- Se mejoró la edición de la búsqueda en las páginas «Traducir».
- Mejoró el tratamiento de actualizaciones de repositorio concurrentes.
- Se incluye el idioma de origen en el formulario de creación de proyecto.

- Se incluye el recuento de cambios en los créditos.
- Se arregló la selección del idioma de interfaz en casos puntuales.
- Allow to whitelist registration methods with registrations closed.
- Se mejoró la búsqueda de términos relacionados en el glosario.
- Improved translation memory matches.
- Se agrupan los resultados idénticos en la automatización.
- Se añadió un enlace directo para editar capturas de pantalla desde la página de traducción.
- Se mejoró el cuadro de diálogo de confirmación ante eliminaciones.
- Se incluyen las plantillas en las descargas ZIP.
- Se permite la utilización de Markdown y la configuración de notificaciones en los anuncios.
- Detalles ampliados en los listados de comprobaciones.
- Se añadió compatibilidad con formatos de archivo nuevos: *Cadenas PHP de Laravel*, *Archivos HTML*, *Formato OpenDocument*, *Formato IDML*, *Archivos RC de Windows*, *Traducciones en INI*, *Traducciones INI de Inno Setup*, *Propiedades GWT*, *Archivos JSON de go-i18n*, *Archivo ARB*.
- Consistently use dismissed as state of dismissed checks.
- Add support for configuring default add-ons to enable.
- Se reparó el atajo de teclado en el editor para pasar por alto las comprobaciones.
- 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.

4.33 Weblate 4.0.4

Released on May 7th 2020.

- Se arregló la ejecución del conjunto de pruebas en determinados entornos con Python 3.8.
- Typo fixes in the documentation.
- Fixed creating components using API in some cases.
- Fixed JavaScript errors breaking mobile navigation.
- Fixed crash on displaying some checks.
- Fixed screenshots listing.
- Fixed monthly digest notifications.
- Fixed intermediate translation behavior with units non existing in translation.

4.34 Weblate 4.0.3

Released on May 2nd 2020.

- Fixed possible crash in reports.
- Las menciones a usuarios en los comentarios ya no distinguen entre mayúsculas y minúsculas.
- Fixed PostgreSQL migration for non superusers.
- Fixed changing the repository URL while creating component.
- Se arregló un cierre inesperado cuando el repositorio de origen ascendente desaparece.

4.35 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.

4.36 Weblate 4.0.1

Publicada el 16 de abril de 2020.

- Se arregló la instalación del paquete desde PyPI.

4.37 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 add-on to handle more syntax variants.
- Fixed editing monolingual XLIFF source file.
- Added support for exact matching in *Búsquedas*.
- Extended API to cover screenshots, users, groups, componentlists and extended creating projects.
- Add support for source upload on bilingual translations.
- Added support for intermediate language from developers.
- Added support for source strings review.
- Extended download options for platform wide translation memory.

4.38 Serie 3.x de Weblate

4.38.1 Weblate 3.11.3

Publicada el 11 de marzo de 2020.

- Se corrigió la búsqueda de campos con una determinada prioridad.
- Se corrigió la consulta predefinida para cadenas añadidas recientemente.
- Se solucionó la duplicación de resultados en la búsqueda.
- Se reparó la representación gráfica de las notificaciones en Gmail.
- Se corrigió la reversión de cambios desde el historial.
- Se añadieron enlaces a sucesos en las notificaciones de resumen.
- Se corrigió la dirección de correo en la confirmación de eliminación de cuenta.
- Se permite la autenticación con Slack en contenedores Docker.
- Se evita enviar notificaciones para idiomas a los que no se ha suscrito.
- Se incluyen las colas de Celery en la vista de conjunto del rendimiento.
- Fixed documentation links for add-ons.
- Reduced false negatives for unchanged translation check.
- Se incrementó la dependencia a bleach para solucionar la vulnerabilidad CVE-2020-6802.
- Fixed listing project level changes in history.
- Fixed stats invalidation in some corner cases.
- Se solucionó la búsqueda de determinados estados de cadenas.
- Improved format string checks behavior on missing percent.
- Se reparó la autenticación a través de determinados proveedores de terceros.

4.38.2 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.

4.38.3 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.

4.38.4 Weblate 3.11

Publicada el 17 de febrero de 2020.

- Allow using VCS push URL during component creation via API.
- Rendered width check now shows image with the render.
- Se repararon los enlaces en los mensajes de notificación por correo electrónico.
- Se mejoró el aspecto de los mensajes de correo de texto sencillo.
- Se muestran permanentemente las comprobaciones ignoradas para permitir activarlas nuevamente.
- Se muestran las claves cercanas en las traducciones monolingües.
- Se permite agrupar las formas de las cadenas.
- Se recomienda actualizar Weblate en las comprobaciones del sistema.
- Provide more detailed analysis for duplicate language alert.
- Se detalló la información sobre licencias en las páginas de proyecto.
- Automatically unshallow local copies if needed.
- Se arregló la descarga de cadenas pendientes de intervención.
- New alert to warn about using the same file mask twice.
- Improve XML placeables extraction.
- The `SINGLE_PROJECT` can now enforce redirection to chosen project.
- Se añadió una opción para marcar comentarios como resueltos.
- Added bulk editing of flags.
- Added support for labels.
- Added bulk edit add-on.
- Added option for *Forzar comprobaciones*.
- Se incrementó la validez predeterminada de los enlaces de confirmación.
- Se mejoró la integración con Matomo.
- Fixed *Se había traducido* to correctly handle source string change.
- Extended automatic updates configuration by `AUTO_UPDATE`.
- LINGUAS add-ons now do full sync of translations in Weblate.

4.38.5 Weblate 3.10.3

Publicada el 18 de abril de 2020.

- Compatibilidad con translate-toolkit 2.5.0.

4.38.6 Weblate 3.10.2

Publicada el 18 de abril de 2020.

- Se añadió un indicador de bloqueo para los proyectos.
- Se arregló un defecto en el CSS que provocaba parpadeos en determinados navegadores web.
- Se repararon las búsquedas en sistemas con configuraciones regionales distintas del inglés.
- Improved repository matching for GitHub and Bitbucket hooks.
- Se corrigió la migración de datos en algunas instalaciones con Python 2.7.
- Allow configuration of Git shallow cloning.
- Se mejoró el procesamiento en segundo plano de las notificaciones.
- Fixed broken form submission when navigating back in web browser.
- New add-on to configure YAML formatting.
- Fixed same plurals check to not fire on single plural form languages.
- Se arregló la búsqueda por expresiones regulares en algunos campos.

4.38.7 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.

4.38.8 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.

4.38.9 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.

4.38.10 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 add-on to bootstrap translations.
- Extend automatic translation to add suggestions.
- Display add-on parameters on overview.
- Sentry is now supported through modern Sentry SDK instead of Raven.
- Changed example settings to be better fit for production environment.
- Added automated backups using BorgBackup.
- Split cleanup add-on for RESX to avoid unwanted file updates.
- Added advanced search capabilities.
- Allow users to download their own reports.
- Added localization guide to help configuring components.
- Added support for GitLab merge requests.
- Improved display of repository status.
- Perform automated translation in the background.

4.38.11 Weblate 3.8

Publicada el 15 de agosto de 2019.

- Added support for simplified creating of similar components.
- Added support for parsing translation flags from the XML based file formats.
- Log exceptions into Celery log.
- Improve performance of repository scoped add-ons.
- Se mejoró la apariencia de los mensajes de notificación por correo.
- Se arregló el comportamiento del restablecimiento de contraseñas.
- Se mejoró el rendimiento en la mayoría de las páginas de traducción.
- Fixed listing of languages not known to Weblate.
- Add support for cloning add-ons to discovered components.
- Add support for replacing file content with uploaded.
- Add support for translating non VCS based content.
- Added OpenGraph widget image to use on social networks.

- Added support for animated screenshots.
- Improved handling of monolingual XLIFF files.
- Avoid sending multiple notifications for single event.
- Add support for filtering changes.
- Extended predefined periods for reporting.
- Added webhook support for Azure Repos.
- New opt-in notifications on pending suggestions or untranslated strings.
- Add one click unsubscribe link to notification e-mails.
- Fixed false positives with Has been translated check.
- New management interface for admins.
- String priority can now be specified using flags.
- Added language management views.
- Add checks for Qt library and Ruby format strings.
- Added configuration to better fit single project installations.
- Notify about new string on source string change on monolingual translations.
- Added separate view for translation memory with search capability.

4.38.12 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.

4.38.13 Weblate 3.7

Publicada el 21 de junio de 2019.

- Added separate Celery queue for notifications.
- Use consistent look with application for API browsing.
- Include approved stats in the reports.
- Report progress when updating translation component.
- Allow to abort running background component update.
- Extend template language for filename manipulations.
- Use templates for editor link and repository browser URL.
- Indicate max length and current characters count when editing translation.
- Improved handling of abbreviations in unchanged translation check.

- Refreshed landing page for new contributors.
- Add support for configuring msgmerge add-on.
- Delay opening SMTP connection when sending notifications.
- Improved error logging.
- Allow custom location in MO generating add-on.
- Added add-ons to cleanup old suggestions or comments.
- Added option to enable horizontal mode in the Zen editor.
- Improved import performance with many linked components.
- Fixed examples installation in some cases.
- Improved rendering of alerts in changes.
- Added new horizontal stats widget.
- Improved format strings check on plurals.
- Added font management tool.
- New check for rendered text dimensions.
- Added support for subtitle formats.
- Include overall completion stats for languages.
- Added reporting at project and global scope.
- Improved user interface when showing translation status.
- New Weblate logo and color scheme.
- New look of bitmap badges.

4.38.14 Weblate 3.6.1

Publicada el 26 de abril de 2019.

- Improved handling of monolingual XLIFF files.
- Fixed digest notifications in some corner cases.
- Fixed add-on script error alert.
- Fixed generating MO file for monolingual PO files.
- Fixed display of uninstalled checks.
- Indicate administered projects on project listing.
- Allow update to recover from missing VCS repository.

4.38.15 Weblate 3.6

Publicada el 20 de abril de 2019.

- Add support for downloading user data.
- Add-ons are now automatically triggered upon installation.
- Improved instructions for resolving merge conflicts.
- Cleanup add-on is now compatible with app store metadata translations.
- Configurable language code syntax when adding new translations.

- Warn about using Python 2 with planned termination of support in April 2020.
- Extract special characters from the source string for visual keyboard.
- Extended contributor stats to reflect both source and target counts.
- Admins and consistency add-ons can now add translations even if disabled for users.
- Fixed description of toggle disabling Language–Team header manipulation.
- Notify users mentioned in comments.
- Removed file format autodetection from component setup.
- Fixed generating MO file for monolingual PO files.
- Added digest notifications.
- Added support for muting component notifications.
- Se añadieron notificaciones al haber nuevas alertas, mensajes en pizarra o componentes.
- Notifications for administered projects can now be configured.
- Improved handling of three letter language codes.

4.38.16 Weblate 3.5.1

Publicada el 10 de marzo de 2019.

- Fixed Celery systemd unit example.
- Fixed notifications from HTTP repositories with login.
- Fixed race condition in editing source string for monolingual translations.
- Include output of failed add-on execution in the logs.
- Improved validation of choices for adding new language.
- Se permite editar el formato de archivo en la configuración del componente.
- Se actualizaron las instrucciones de instalación para preferir Python 3.
- Mejoras de rendimiento y coherencia durante la carga de traducciones.
- El servicio de terminología de Microsoft se hizo compatible con las versiones actuales de Zeep.
- Actualización de las regionalizaciones.

4.38.17 Weblate 3.5

Publicada el 3 de marzo de 2019.

- Se mejoró el rendimiento de la memoria de traducción incorporada.
- Se añadió una interfaz para gestionar la memoria de traducción global.
- Improved alerting on bad component state.
- Se añadió una interfaz para gestionar los mensajes de pizarra.
- Add-on commit message now can be configured.
- Reduce number of commits when updating upstream repository.
- Fixed possible metadata loss when moving component between projects.
- Se perfeccionó la navegación en el modo zen.
- Se sumaron varias comprobaciones de calidad nuevas (relativas a Markdown y URL).

- Se añadió compatibilidad para archivos de metadatos de tiendas de aplicaciones.
- Es posible ahora activar o desactivar las integraciones con GitHub y Gerrit.
- Se añadió una comprobación de caracteres «kashida».
- Added option to squash commits based on authors.
- Se mejoró la compatibilidad con el formato de archivo XLSX.
- Compatibilidad con Tesseract 4.0.
- Billing add-on now removes projects for unpaid billings after 45 days.

4.38.18 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 add-on to squash commits prior to push.
- Se mejoró la visualización de los cambios a la traducción.
- Changed default merge style to rebase and made that configurable.
- Better handle private use subtags in language code.
- Se mejoró el rendimiento de las actualizaciones de índice de texto completo.
- Extended file upload API to support more parameters.

4.38.19 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.

4.38.20 Weblate 3.2.2

Publicada el 20 de octubre de 2018.

- Remove no longer needed Babel dependency.
- Updated language definitions.
- Improve documentation for add-ons, LDAP and Celery.
- Fixed enabling new dos-eol and auto-java-messageformat flags.
- Fixed running setup.py test from PyPI package.
- Improved plurals handling.
- Fixed translation upload API failure in some corner cases.
- Fixed updating Git configuration in case it was changed manually.

4.38.21 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.

4.38.22 Weblate 3.2

Publicada el 6 de octubre de 2018.

- Add install_addon management command for automated add-on installation.
- Allow more fine grained ratelimit settings.
- Added support for export and import of Excel files.
- Improve component cleanup in case of multiple component discovery add-ons.
- Rewritten Microsoft Terminology machine translation backend.
- Weblate now uses Celery to offload some processing.
- Improved search capabilities and added regular expression search.
- Added support for Youdao Zhiyun API machine translation.
- Added support for Baidu API machine translation.
- Integrated maintenance and cleanup tasks using Celery.
- Improved performance of loading translations by almost 25%.
- Removed support for merging headers on upload.
- Removed support for custom commit messages.

- Configurable editing mode (zen/full).
- Added support for error reporting to Sentry.
- Added support for automated daily update of repositories.
- Added support for creating projects and components by users.
- Built-in translation memory now automatically stores translations done.
- Users and projects can import their existing translation memories.
- Better management of related strings for screenshots.
- Added support for checking Java MessageFormat.

See [3.2 milestone on GitHub](#) for detailed list of addressed issues.

4.38.23 Weblate 3.1.1

Publicada el 27 de julio de 2018.

- Fix testsuite failure on some setups.

4.38.24 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 libavatar service.
- New add-on 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.

4.38.25 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.

4.38.26 Weblate 3.0

Publicada el 1.º de junio de 2018.

- Rewritten access control.
- Several code cleanups that lead to moved and renamed modules.
- New add-on for automatic component discovery.
- The `import_project` management command has now slightly different parameters.
- Added basic support for Windows RC files.
- New add-on to store contributor names in PO file headers.
- The per component hook scripts are removed, use add-ons instead.
- Add support for collecting contributor agreements.
- Access control changes are now tracked in history.
- New add-on to ensure all components in a project have same translations.
- Support for more variables in commit message templates.
- Add support for providing additional textual context.

4.39 Serie 2.x de Weblate

4.39.1 Weblate 2.20

Publicada el 4 de abril de 2018.

- Improved speed of cloning subversion repositories.
- Changed repository locking to use third party library.
- Added support for downloading only strings needing action.
- Se permite efectuar búsquedas en varios idiomas a la vez.
- New add-on to configure gettext output wrapping.
- New add-on to configure JSON formatting.
- Added support for authentication in API using RFC 6750 compatible Bearer authentication.
- Added support for automatic translation using machine translation services.
- Se admite código HTML en los mensajes de la pizarra.
- Added support for mass changing state of strings.
- Translate-toolkit at least 2.3.0 is now required, older versions are no longer supported.
- 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.

4.39.2 Weblate 2.19.1

Publicada el 20 de febrero de 2018.

- Fixed migration issue on upgrade from 2.18.
- Improved file upload API validation.

4.39.3 Weblate 2.19

Publicada el 15 de febrero de 2018.

- Fixed imports across some file formats.
- Display human friendly browser information in audit log.
- Added TMX exporter for files.
- Various performance improvements for loading translation files.
- Added option to disable access management in Weblate in favor of Django one.
- Improved glossary lookup speed for large strings.
- Compatibility with django_auth_ldap 1.3.0.
- Configuration errors are now stored and reported persistently.
- Honor ignore flags in whitespace autofixer.
- Improved compatibility with some Subversion setups.
- Improved built-in machine translation service.
- Added support for SAP Translation Hub service.
- Added support for Microsoft Terminology service.
- Removed support for advertisement in notification e-mails.
- Improved translation progress reporting at language level.
- Improved support for different plural formulas.
- Added support for Subversion repositories not using stdlayout.
- Added add-ons to customize translation workflows.

4.39.4 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.

4.39.5 Weblate 2.17.1

Publicada el 13 de octubre de 2017.

- Fixed running testsuite in some specific situations.
- Locales updates.

4.39.6 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.

4.39.7 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.

4.39.8 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.

4.39.9 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.

4.39.10 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.

4.39.11 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.

4.39.12 Weblate 2.13

Publicada el 12 de abril de 2017.

- Fixed quality checks on translation templates.
- Se añadió una comprobación de calidad que se desencadena al perder traducciones.
- Se añadió una opción para ver las sugerencias pendientes de un usuario.
- Se añadió una opción para crear listas de componentes automáticamente.
- Es posible configurar el cuadro de mando que ven de manera predeterminada los usuarios no autenticados.
- Se añadió una opción para examinar 25 cadenas al azar para su revisión.
- El historial ahora incluye los cambios a las cadenas.
- Better error reporting when adding new translation.
- Added per language search within project.
- Group ACLs can now be limited to certain permissions.
- The per project ACLs are now implemented using Group ACL.
- Added more fine grained privileges control.
- Various minor UI improvements.

4.39.13 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.

4.39.14 Weblate 2.11

Publicada el 31 de enero de 2017.

- Include language detailed information on language page.
- Mercurial backend improvements.
- Added option to specify translation component priority.
- More consistent usage of Group ACL even with less used permissions.
- Added WL_BRANCH variable to hook scripts.
- Improved developer documentation.
- Better compatibility with various Git versions in Git exporter add-on.
- Included per project and component stats.
- Added language code mapping for better support of Microsoft Translate API.
- Moved fulltext cleanup to background job to make translation removal faster.
- Fixed displaying of plural source for languages with single plural form.
- Improved error handling in import_project.
- Various performance improvements.

4.39.15 Weblate 2.10.1

Publicada el 20 de enero de 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

4.39.16 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.

4.39.17 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.

4.39.18 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.

4.39.19 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.

4.39.20 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.

4.39.21 Weblate 2.5

Publicada el 10 de marzo de 2016.

- Fixed automatic translation for project owners.
- Improved performance of commit and push operations.
- New management command to add suggestions from command-line.
- Added support for merging comments on file upload.
- Added support for some GNU extensions to C printf format.
- Documentation improvements.
- Added support for generating translator credits.
- Added support for generating contributor stats.
- Site wide search can search only in one language.
- Improve quality checks for Armenian.
- Support for starting translation components without existing translations.
- Support for adding new translations in Qt TS.
- Improved support for translating PHP files.
- Mejoras de rendimiento para las comprobaciones de calidad.
- Se arregló la búsqueda en todo el sitio de comprobaciones fallidas.

- Se añadió una opción para especificar el idioma de origen.
- Improved support for XLIFF files.
- Se amplió la lista de opciones disponibles para `import_project`.
- Improved targeting for whiteboard messages.
- Es posible efectuar traducciones automáticas en diversos proyectos a la vez.
- Se optimizó la indización de la búsqueda de texto completo.
- Se añadió una orden de gestión para traducciones automáticas.
- Added placeables highlighting.
- Added keyboard shortcuts for placeables, checks and machine translations.
- Improved translation locking.
- Added quality check for AngularJS interpolation.
- Added extensive group based ACLs.
- Clarified terminology on strings needing edit (formerly fuzzy).
- Clarified terminology on strings needing action and untranslated strings.
- Se admite Python 3.
- Ya no se admite Django 1.7.
- Dropped dependency on msginit for creating new gettext PO files.
- Added configurable dashboard views.
- Improved notifications on parse errors.
- Added option to import components with duplicate name to `import_project`.
- Improved support for translating PHP files.
- Added XLIFF export for dictionary.
- Added XLIFF and gettext PO export for all translations.
- Documentation improvements.
- Added support for configurable automatic group assignments.
- Improved adding of new translations.

4.39.22 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.

4.39.23 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.

4.39.24 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.

4.39.25 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.

4.39.26 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.

4.40 Serie 1.x de Weblate

4.40.1 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.

4.40.2 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.

4.40.3 Weblate 1.7

Publicada el 7 de octubre de 2013.

- Please check manual for upgrade instructions.
- Support for checking Python brace format string.
- Per component customization of quality checks.
- Detailed per translation stats.
- Changed way of linking suggestions, checks and comments to strings.
- Users can now add text to commit message.
- Support for subscribing on new language requests.
- Support for adding new translations.
- Widgets and charts are now rendered using Pillow instead of Pango + Cairo.
- Add status badge widget.
- Dropped invalid text direction check.
- Los cambios realizados a los diccionarios ahora figuran en el historial.
- Performance improvements for translation view.

4.40.4 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.

4.40.5 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.

4.40.6 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.

4.40.7 Weblate 1.3

Publicada el 16 de noviembre de 2012.

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Improved quality checks processing.
- Added new checks (BBCode, XML markup and newlines).
- Support for optional rebasing instead of merge.
- Possibility to relocate Weblate (for example to run it under /weblate path).
- Support for manually choosing file type in case autodetection fails.
- Better support for Android resources.
- Support for generating SSH key from web interface.
- More visible data exports.
- New buttons to enter some special characters.
- Support for exporting dictionary.
- Support for locking down whole Weblate installation.
- Checks for source strings and support for source strings review.
- Support for user comments for both translations and source strings.
- Better changes log tracking.
- Changes can now be monitored using RSS.
- Improved support for RTL languages.

4.40.8 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.

4.40.9 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.

4.40.10 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.

4.41 Serie 0.x de Weblate

4.41.1 Weblate 0.9

Publicada el 18 de abril de 2012.

- Se arregló la importación de idiomas desconocidos.
- Improved listing of nearby messages.
- Se mejoraron varias comprobaciones.
- Puesta al día de la documentación.

- Added definition for several more languages.
- Various code cleanups.
- Documentation improvements.
- Changed file layout.
- Update helper scripts to Django 1.4.
- Improved navigation while translating.
- Better handling of po file renames.
- Better validation while creating component.
- Integrated full setup into syncdb.
- Added list of recent changes to all translation pages.
- Check for untranslated strings ignores format string only messages.

4.41.2 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.

4.41.3 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.

4.41.4 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.

4.41.5 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).

4.41.6 Weblate 0.4

Publicada el 8 de febrero de 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

4.41.7 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.

4.41.8 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.

4.41.9 Weblate 0.1

Publicada el 6 de febrero de 2012.

- Versión inicial.

W

wlc, [148](#)
wlc.config, [148](#)
wlc.main, [149](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component:slug)/, 124
ANY /, 96	GET /api/components/(string:project)/(string:component:slug)/, 120
/api	GET /api/components/(string:project)/(string:component:slug)/, 122
GET /api/, 99	GET /api/components/(string:project)/(string:component:slug)/, 122
/api/addons	GET /api/components/(string:project)/(string:component:slug)/, 121
GET /api/addons/, 136	GET /api/components/(string:project)/(string:component:slug)/, 120
GET /api/addons/(int:id)/, 136	GET /api/components/(string:project)/(string:component:slug)/, 124
PUT /api/addons/(int:id)/, 136	GET /api/components/(string:project)/(string:component:slug)/, 122
DELETE /api/addons/(int:id)/, 137	POST /api/components/(string:project)/(string:component:slug)/, 136
PATCH /api/addons/(int:id)/, 136	POST /api/components/(string:project)/(string:component:slug)/, 124
/api/changes	POST /api/components/(string:project)/(string:component:slug)/, 120
GET /api/changes/, 133	POST /api/components/(string:project)/(string:component:slug)/, 121
GET /api/changes/(int:id)/, 133	POST /api/components/(string:project)/(string:component:slug)/, 123
/api/component-lists	PUT /api/components/(string:project)/(string:component:slug)/, 119
GET /api/component-lists/, 137	DELETE /api/components/(string:project)/(string:component:slug)/, 119
GET /api/component-lists/(str:slug)/, 137	DELETE /api/components/(string:project)/(string:component:slug)/, 125
POST /api/component-lists/(str:slug)/components/, 138	PATCH /api/components/(string:project)/(string:component:slug)/, 117
PUT /api/component-lists/(str:slug)/, 137	/api/groups
DELETE /api/component-lists/(str:slug)/, 137	GET /api/groups/, 103
DELETE /api/component-lists/(str:slug)/components/(string:component:slug)/, 138	GET /api/groups/(int:id)/, 103
PATCH /api/component-lists/(str:slug)/, 137	POST /api/groups/, 103
/api/components	POST /api/groups/(int:id)/changes/, 106
GET /api/components/, 115	POST /api/groups/(int:id)/components/, 105
GET /api/components/(string:project)/(string:component:slug)/, 115	
GET /api/components/(string:project)/(string:component:slug)/, 119	
GET /api/components/(string:project)/(string:component:slug)/, 120	

POST /api/groups/(int:id)/languages/, 105
 PATCH /api/projects/(string:project)/, 110

POST /api/groups/(int:id)/projects/, 105

POST /api/groups/(int:id)/roles/, 105

PUT /api/groups/(int:id)/, 104

DELETE /api/groups/(int:id)/, 104

DELETE /api/groups/(int:id)/componentlists/(int:componentlist_id)/, 106

DELETE /api/groups/(int:id)/components/(int:component_id)/, 105

DELETE /api/groups/(int:id)/languages/(string:language)/, 105

DELETE /api/groups/(int:id)/projects/(int:project_id)/, 105

PATCH /api/groups/(int:id)/, 104

/api/languages

GET /api/languages/, 107

GET /api/languages/(string:language)/, 107

GET /api/languages/(string:language)/statistics/, 108

POST /api/languages/, 107

PUT /api/languages/(string:language)/, 108

DELETE /api/languages/(string:language)/, 108

PATCH /api/languages/(string:language)/, 108

/api/metrics

GET /api/metrics/, 139

/api/projects

GET /api/projects/, 109

GET /api/projects/(string:project)/, 109

GET /api/projects/(string:project)/changes/, 110

GET /api/projects/(string:project)/components/, 112

GET /api/projects/(string:project)/languages/, 114

GET /api/projects/(string:project)/repository/, 110

GET /api/projects/(string:project)/statistics/, 115

POST /api/projects/, 109

POST /api/projects/(string:project)/components/, 112

POST /api/projects/(string:project)/repository/, 111

PUT /api/projects/(string:project)/, 110

DELETE /api/projects/(string:project)/, 110

/api/roles

GET /api/roles/, 106

GET /api/roles/(int:id)/, 106

POST /api/roles/, 106

DELETE /api/roles/(int:id)/, 107

PATCH /api/roles/(int:id)/, 107

/api/screenshots

GET /api/screenshots/, 133

GET /api/screenshots/(int:id)/, 133

GET /api/screenshots/(int:id)/file/, 134

POST /api/screenshots/, 135

POST /api/screenshots/(int:id)/file/, 134

POST /api/screenshots/(int:id)/units/, 134

DELETE /api/screenshots/(int:id)/, 135

DELETE /api/screenshots/(int:id)/units/(int:unit_id)/, 134

PATCH /api/screenshots/(int:id)/, 135

/api/tasks

GET /api/tasks/, 138

GET /api/tasks/(str:uuid)/, 138

/api/translations

GET /api/translations/, 125

GET /api/translations/(string:project)/(string:component)/, 125

GET /api/translations/(string:project)/(string:component)/, 127

GET /api/translations/(string:project)/(string:component)/, 129

GET /api/translations/(string:project)/(string:component)/, 129

GET /api/translations/(string:project)/(string:component)/, 129

GET /api/translations/(string:project)/(string:component)/, 128

POST /api/translations/(string:project)/(string:component)/, 129

POST /api/translations/(string:project)/(string:component)/, 130

POST /api/translations/(string:project)/(string:component)/, 128

DELETE /api/translations/(string:project)/(string:component)/, 127

/api/units

GET /api/units/, 131

GET /api/units/(int:id)/, 131
PUT /api/units/(int:id)/, 132
DELETE /api/units/(int:id)/, 132
PATCH /api/units/(int:id)/, 132

/api/users

GET /api/users/, 100
GET /api/users/(str:username)/, 100
GET /api/users/(str:username)/notifications/,
102
GET /api/users/(str:username)/notifications/(int:subscription_id)/,
102
GET /api/users/(str:username)/statistics/,
101
POST /api/users/, 100
POST /api/users/(str:username)/groups/,
101
POST /api/users/(str:username)/notifications/,
102
PUT /api/users/(str:username)/, 101
PUT /api/users/(str:username)/notifications/(int:subscription_id)/,
102
DELETE /api/users/(str:username)/, 101
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/,
103
PATCH /api/users/(str:username)/, 101
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/,
102

/exports

GET /exports/rss/, 142
GET /exports/rss/(string:project)/, 142
GET /exports/rss/(string:project)/(string:component)/,
142
GET /exports/rss/(string:project)/(string:component)/(string:language)/,
142
GET /exports/rss/language/(string:language)/,
142
GET /exports/stats/(string:project)/(string:component)/,
141

/hooks

GET /hooks/update/(string:project)/,
139
GET /hooks/update/(string:project)/(string:component)/,
139
POST /hooks/azure/, 140
POST /hooks/bitbucket/, 140
POST /hooks/gitea/, 140
POST /hooks/gitee/, 141
POST /hooks/github/, 139
POST /hooks/gitlab/, 140
POST /hooks/pagure/, 140

Símbolos

- .XML resource file
 - file format, [81](#)
- add
 - auto_translateopción de línea de comando, [369](#)
- addon
 - install_addonopción de línea de comando, [376](#)
- age
 - commit_pendingopción de línea de comando, [370](#)
- author
 - add_suggestionsopción de línea de comando, [369](#)
- author-email
 - wlcoopción de línea de comando, [146](#)
- author-name
 - wlcoopción de línea de comando, [146](#)
- base-file-template
 - import_projectopción de línea de comando, [373](#)
- check
 - import_usersopción de línea de comando, [375](#)
- config
 - wlcoopción de línea de comando, [144](#)
- config-section
 - wlcoopción de línea de comando, [144](#)
- configuration
 - install_addonopción de línea de comando, [376](#)
- convert
 - wlcoopción de línea de comando, [145](#)
- email
 - create_adminopción de línea de comando, [371](#)
- file-format
 - import_projectopción de línea de comando, [374](#)
- force
 - load_poopción de línea de comando, [377](#)
- force-commit
 - pushgitopción de línea de comando, [378](#)
- format
 - wlcoopción de línea de comando, [144](#)
- fuzzy
 - wlcoopción de línea de comando, [146](#)
- ignore
 - import_jsonopción de línea de comando, [372](#)
- inconsistent
 - auto_translateopción de línea de comando, [369](#)
- input
 - wlcoopción de línea de comando, [146](#)
- key
 - wlcoopción de línea de comando, [144](#)
- lang
 - load_poopción de línea de comando, [377](#)
- language-code
 - list_translatorsopción de línea de comando, [376](#)
- language-map
 - import_memoryopción de línea de comando, [373](#)
- language-regex
 - import_projectopción de línea de comando, [374](#)
- license
 - import_projectopción de línea de comando, [374](#)
- license-url
 - import_projectopción de línea de comando, [374](#)
- main-component
 - import_jsonopción de línea de comando, [372](#)
 - import_projectopción de línea de comando, [374](#)
- method
 - wlcoopción de línea de comando, [146](#)
- mode
 - auto_translateopción de línea de

comando, [369](#)

--mt
auto_translateopción de línea de comando, [369](#)

--name
createadminopción de línea de comando, [371](#)

--name-template
import_projectopción de línea de comando, [373](#)

--new-base-template
import_projectopción de línea de comando, [374](#)

--no-password
createadminopción de línea de comando, [371](#)

--no-privs-update
setupgroupsopción de línea de comando, [378](#)

--no-projects-update
setupgroupsopción de línea de comando, [378](#)

--no-update
setuplangopción de línea de comando, [379](#)

--output
wlcopción de línea de comando, [145](#)

--overwrite
auto_translateopción de línea de comando, [369](#)
wlcopción de línea de comando, [145](#)

--password
createadminopción de línea de comando, [371](#)

--project
import_jsonopción de línea de comando, [372](#)

--source
auto_translateopción de línea de comando, [369](#)

--threshold
auto_translateopción de línea de comando, [369](#)

--update
createadminopción de línea de comando, [371](#)
import_jsonopción de línea de comando, [372](#)
install_addonopción de línea de comando, [376](#)

--url
wlcopción de línea de comando, [144](#)

--user
auto_translateopción de línea de comando, [369](#)

--username
createadminopción de línea de comando, [371](#)

--vcs
import_projectopción de línea de comando, [374](#)

A

add_suggestions
weblate admin command, [369](#)

add_suggestionsopción de línea de comando
--author, [369](#)

ADMINS
setting, [197](#)

AKISMET_API_KEY
setting, [321](#)

ALLOWED_HOSTS
setting, [197](#)

Android
file format, [76](#)

ANONYMOUS_USER_NAME
setting, [321](#)

API, [96](#), [142](#), [147](#)

Apple strings
file format, [77](#)

ARB
file format, [80](#)

AUDITLOG_EXPIRY
setting, [322](#)

AUTH_LOCK_ATTEMPTS
setting, [322](#)

AUTH_TOKEN_VALID
setting, [323](#)

auto_translate
weblate admin command, [369](#)

auto_translateopción de línea de comando
--add, [369](#)
--inconsistent, [369](#)
--mode, [369](#)
--mt, [369](#)
--overwrite, [369](#)
--source, [369](#)
--threshold, [369](#)
--user, [369](#)

AUTO_UPDATE
setting, [322](#)

AUTOFIX_LIST
setting, [323](#)

AVATAR_URL_PREFIX
setting, [322](#)

B

BACKGROUND_TASKS
setting, [324](#)

BASE_DIR
setting, [324](#)

BaseAddon (*clase en weblate.addons.base*), [413](#)

BASIC_LANGUAGES
setting, [324](#)

bilingual
translation, 68
BORG_EXTRA_ARGS
setting, 324

C

can_install() (*método de clase de weblate.addons.base.BaseAddon*), 413
CELERY_BACKUP_OPTIONS, 155, 172
CELERY_BEAT_OPTIONS, 155, 172
CELERY_MAIN_OPTIONS, 155, 172
CELERY_MEMORY_OPTIONS, 155, 172
CELERY_NOTIFY_OPTIONS, 155, 172
celery_queues
weblate admin command, 370
CELERY_TRANSLATE_OPTIONS, 155, 172
changes
wlcopción de línea de comando, 145
CHECK_LIST
setting, 325
checkgit
weblate admin command, 370
cleanup
wlcopción de línea de comando, 145
cleanup_ssh_keys
weblate admin command, 371
cleanuptrans
weblate admin command, 370
Comma separated values
file format, 82
Command (*clase en wlc.main*), 149
COMMENT_CLEANUP_DAYS
setting, 326
commit
wlcopción de línea de comando, 144
commit_pending
weblate admin command, 370
COMMIT_PENDING_HOURS
setting, 326
commit_pendingopción de línea de comando
--age, 370
commitgit
weblate admin command, 370
configure() (*método de weblate.addons.base.BaseAddon*), 413
CONTACT_FORM
setting, 326
createadmin
weblate admin command, 371
createadminopción de línea de comando
--email, 371
--name, 371
--no-password, 371
--password, 371
--update, 371
--username, 371
CSP_CONNECT_SRC

setting, 325
CSP_FONT_SRC
setting, 325
CSP_IMG_SRC
setting, 325
CSP_SCRIPT_SRC
setting, 325
CSP_STYLE_SRC
setting, 325
CSV
file format, 82

D

daily() (*método de weblate.addons.base.BaseAddon*), 413
DATA_DIR
setting, 326
DATABASE_BACKUP
setting, 327
DATABASES
setting, 198
DEBUG
setting, 198
DEFAULT_ACCESS_CONTROL
setting, 327
DEFAULT_ADD_MESSAGE
setting, 328
DEFAULT_ADDON_MESSAGE
setting, 328
DEFAULT_ADDONS
setting, 328
DEFAULT_AUTO_WATCH
setting, 327
DEFAULT_COMMIT_MESSAGE
setting, 328
DEFAULT_COMMITER_EMAIL
setting, 328
DEFAULT_COMMITER_NAME
setting, 328
DEFAULT_DELETE_MESSAGE
setting, 328
DEFAULT_FROM_EMAIL
setting, 198
DEFAULT_LANGUAGE
setting, 329
DEFAULT_MERGE_MESSAGE
setting, 328
DEFAULT_MERGE_STYLE
setting, 329
DEFAULT_PAGE_LIMIT
setting, 344
DEFAULT_PULL_MESSAGE
setting, 329
DEFAULT_RESTRICTED_COMPONENT
setting, 327
DEFAULT_SHARED_TM
setting, 329
DEFAULT_TRANSLATION_PROPAGATION

- setting, 329
- download
 - wlcopción de línea de comando, 145
- DTD
 - file format, 84
- dump_memory
 - weblate admin command, 371
- dumpuserdata
 - weblate admin command, 372

E

- ENABLE_AVATARS
 - setting, 330
- ENABLE_HOOKS
 - setting, 330
- ENABLE_HTTPS
 - setting, 330
- ENABLE_SHARING
 - setting, 330

F

- file format
 - .XML resource file, 81
 - Android, 76
 - Apple strings, 77
 - ARB, 80
 - Comma separated values, 82
 - CSV, 82
 - DTD, 84
 - gettext, 70
 - go-i18n, 80
 - GWT properties, 74
 - i18next, 79
 - INI translations, 75
 - Java properties, 73
 - Joomla translations, 76
 - JSON, 78
 - mi18n lang, 74
 - PHP strings, 77
 - PO, 70
 - Qt, 76
 - RC, 84
 - RESX, 81
 - Ruby YAML, 83
 - Ruby YAML Ain't Markup Language, 83
 - string resources, 76
 - TS, 76
 - XLIFF, 72
 - XML, 84
 - YAML, 83
 - YAML Ain't Markup Language, 83

G

- get() (método de wlc.Weblate), 148
- get_add_form() (método de clase de weblate.addons.base.BaseAddon), 413
- GET_HELP_URL
 - setting, 330

- get_settings_form() (método de weblate.addons.base.BaseAddon), 413
- gettext
 - file format, 70
- GITEA_CREDENTIALS
 - setting, 330
- GITEA_TOKEN
 - setting, 331
- GITEA_USERNAME
 - setting, 331
- GITHUB_CREDENTIALS
 - setting, 332
- GITHUB_TOKEN
 - setting, 332
- GITHUB_USERNAME
 - setting, 332
- GITLAB_CREDENTIALS
 - setting, 331
- GITLAB_TOKEN
 - setting, 332
- GITLAB_USERNAME
 - setting, 332
- go-i18n
 - file format, 80
- GOOGLE_ANALYTICS_ID
 - setting, 333
- GWT properties
 - file format, 74

H

- HIDE_REPO_CREDENTIALS
 - setting, 333
- HIDE_VERSION
 - setting, 333

I

- i18next
 - file format, 79
- import_demo
 - weblate admin command, 372
- import_json
 - weblate admin command, 372
- import_jsonopción de línea de comando
 - ignore, 372
 - main-component, 372
 - project, 372
 - update, 372
- import_memory
 - weblate admin command, 373
- import_memoryopción de línea de comando
 - language-map, 373
- import_project
 - weblate admin command, 373
- import_projectopción de línea de comando
 - base-file-template, 373
 - file-format, 374

- `--language-regex`, 374
- `--license`, 374
- `--license-url`, 374
- `--main-component`, 374
- `--name-template`, 373
- `--new-base-template`, 374
- `--vcs`, 374
- `importuserdata`
 - `weblate admin command`, 375
- `importusers`
 - `weblate admin command`, 375
- `importusers`opción de línea de comando
 - `--check`, 375
- INI translations
 - file format, 75
- `install_addon`
 - `weblate admin command`, 376
- `install_addon`opción de línea de comando
 - `--addon`, 376
 - `--configuration`, 376
 - `--update`, 376
- INTERLEDGER_PAYMENT_POINTERS
 - setting, 333
- IP_BEHIND_REVERSE_PROXY
 - setting, 333
- IP_PROXY_HEADER
 - setting, 334
- IP_PROXY_OFFSET
 - setting, 334
- iPad
 - translation, 77
- iPhone
 - translation, 77
- J**
- Java properties
 - file format, 73
- Joomla translations
 - file format, 76
- JSON
 - file format, 78
- L**
- LEGAL_URL
 - setting, 334
- LICENSE_EXTRA
 - setting, 335
- LICENSE_FILTER
 - setting, 335
- LICENSE_REQUIRED
 - setting, 335
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
 - setting, 336
- `list_languages`
 - `weblate admin command`, 376
- `list_translators`
 - `weblate admin command`, 376
- `list_translators`opción de línea de comando
 - `--language-code`, 376
- `list_versions`
 - `weblate admin command`, 377
- `list-components`
 - wlopción de línea de comando, 144
- `list-languages`
 - wlopción de línea de comando, 144
- `list-projects`
 - wlopción de línea de comando, 144
- `list-translations`
 - wlopción de línea de comando, 144
- `load()` (*método de `wlc.config.WeblateConfig`*), 148
- `loadpo`
 - `weblate admin command`, 377
- `loadpo`opción de línea de comando
 - `--force`, 377
 - `--lang`, 377
- LOCALIZE_CDN_PATH
 - setting, 336
- LOCALIZE_CDN_URL
 - setting, 336
- lock
 - wlopción de línea de comando, 145
- `lock_translation`
 - `weblate admin command`, 377
- lock-status
 - wlopción de línea de comando, 145
- LOGIN_REQUIRED_URLS
 - setting, 336
- LOGIN_REQUIRED_URLS_EXCEPTIONS
 - setting, 336
- ls
 - wlopción de línea de comando, 144
- M**
- MACHINE_TRANSLATION_SERVICES
 - setting, 337
- `main()` (*en el módulo `wlc.main`*), 149
- MATOMO_SITE_ID
 - setting, 337
- MATOMO_URL
 - setting, 337
- mi18n lang
 - file format, 74
- monolingual
 - translation, 68
- `move_language`
 - `weblate admin command`, 377
- MT_APERTIUM_APY
 - setting, 338
- MT_AWS_ACCESS_KEY_ID
 - setting, 338
- MT_AWS_REGION
 - setting, 338
- MT_AWS_SECRET_ACCESS_KEY
 - setting, 338

MT_BAIDU_ID
 setting, [339](#)

MT_BAIDU_SECRET
 setting, [339](#)

MT_DEEPL_API_URL
 setting, [339](#)

MT_DEEPL_KEY
 setting, [339](#)

MT_GOOGLE_CREDENTIALS
 setting, [340](#)

MT_GOOGLE_KEY
 setting, [340](#)

MT_GOOGLE_LOCATION
 setting, [340](#)

MT_GOOGLE_PROJECT
 setting, [340](#)

MT_LIBRETRANSLATE_API_URL
 setting, [340](#)

MT_LIBRETRANSLATE_KEY
 setting, [340](#)

MT_MICROSOFT_BASE_URL
 setting, [341](#)

MT_MICROSOFT_COGNITIVE_KEY
 setting, [341](#)

MT_MICROSOFT_ENDPOINT_URL
 setting, [341](#)

MT_MICROSOFT_REGION
 setting, [341](#)

MT_MODERNMT_KEY
 setting, [341](#)

MT_MODERNMT_URL
 setting, [341](#)

MT_MYMEMORY_EMAIL
 setting, [342](#)

MT_MYMEMORY_KEY
 setting, [342](#)

MT_MYMEMORY_USER
 setting, [342](#)

MT_NETEASE_KEY
 setting, [342](#)

MT_NETEASE_SECRET
 setting, [342](#)

MT_SAP_BASE_URL
 setting, [343](#)

MT_SAP_PASSWORD
 setting, [343](#)

MT_SAP_SANDBOX_APIKEY
 setting, [343](#)

MT_SAP_USE_MT
 setting, [343](#)

MT_SAP_USERNAME
 setting, [343](#)

MT_SERVICES
 setting, [337](#)

MT_TMSERVER
 setting, [342](#)

MT_YANDEX_KEY
 setting, [342](#)

MT_YOUDAO_ID
 setting, [343](#)

MT_YOUDAO_SECRET
 setting, [343](#)

módulo
 wlc, [148](#)
 wlc.config, [148](#)
 wlc.main, [149](#)

N

NEARBY_MESSAGES
 setting, [344](#)

P

PAGURE_CREDENTIALS
 setting, [344](#)

PAGURE_TOKEN
 setting, [344](#)

PAGURE_USERNAME
 setting, [344](#)

PHP strings
 file format, [77](#)

PIWIK_SITE_ID
 setting, [337](#)

PIWIK_URL
 setting, [337](#)

PO
 file format, [70](#)

post() (método de wlc.Weblate), [148](#)

post_add() (método de *weblate.addons.base.BaseAddon*), [413](#)

post_commit() (método de *weblate.addons.base.BaseAddon*), [413](#)

post_push() (método de *weblate.addons.base.BaseAddon*), [413](#)

post_update() (método de *weblate.addons.base.BaseAddon*), [413](#)

pre_commit() (método de *weblate.addons.base.BaseAddon*), [414](#)

pre_push() (método de *weblate.addons.base.BaseAddon*), [414](#)

pre_update() (método de *weblate.addons.base.BaseAddon*), [414](#)

PRIVACY_URL
 setting, [345](#)

pull
 wlcopción de línea de comando, [145](#)

push
 wlcopción de línea de comando, [145](#)

pushgit
 weblate admin command, [378](#)

pushgitopción de línea de comando
 --force-commit, [378](#)

Python, [147](#)

Python Enhancement Proposals
 PEP 484, [410](#), [423](#)

Q

Qt

file format, 76

R

RATELIMIT_ATTEMPTS

setting, 345

RATELIMIT_LOCKOUT

setting, 345

RATELIMIT_WINDOW

setting, 345

RC

file format, 84

register_command() (en el módulo *wlc.main*), 149

REGISTRATION_ALLOW_BACKENDS

setting, 346

REGISTRATION_CAPTCHA

setting, 346

REGISTRATION_EMAIL_MATCH

setting, 346

REGISTRATION_OPEN

setting, 346

repo

wlcopción de línea de comando, 145

REPOSITORY_ALERT_THRESHOLD

setting, 347

REQUIRE_LOGIN

setting, 347

reset

wlcopción de línea de comando, 145

REST, 96

RESX

file format, 81

RFC

RFC 5646, 67

Ruby YAML

file format, 83

Ruby YAML Ain't Markup Language

file format, 83

S

save_state() (método de *weblate.addons.base.BaseAddon*), 414

SECRET_KEY

setting, 198

SENTRY_DSN

setting, 347

SERVER_EMAIL

setting, 198

SESSION_COOKIE_AGE_AUTHENTICATED

setting, 347

SESSION_ENGINE

setting, 198

setting

ADMINS, 197

AKISMET_API_KEY, 321

ALLOWED_HOSTS, 197

ANONYMOUS_USER_NAME, 321

AUDITLOG_EXPIRY, 322

AUTH_LOCK_ATTEMPTS, 322

AUTH_TOKEN_VALID, 323

AUTO_UPDATE, 322

AUTOFIX_LIST, 323

AVATAR_URL_PREFIX, 322

BACKGROUND_TASKS, 324

BASE_DIR, 324

BASIC_LANGUAGES, 324

BORG_EXTRA_ARGS, 324

CHECK_LIST, 325

COMMENT_CLEANUP_DAYS, 326

COMMIT_PENDING_HOURS, 326

CONTACT_FORM, 326

CSP_CONNECT_SRC, 325

CSP_FONT_SRC, 325

CSP_IMG_SRC, 325

CSP_SCRIPT_SRC, 325

CSP_STYLE_SRC, 325

DATA_DIR, 326

DATABASE_BACKUP, 327

DATABASES, 198

DEBUG, 198

DEFAULT_ACCESS_CONTROL, 327

DEFAULT_ADD_MESSAGE, 328

DEFAULT_ADDON_MESSAGE, 328

DEFAULT_ADDONS, 328

DEFAULT_AUTO_WATCH, 327

DEFAULT_COMMIT_MESSAGE, 328

DEFAULT_COMMITTER_EMAIL, 328

DEFAULT_COMMITTER_NAME, 328

DEFAULT_DELETE_MESSAGE, 328

DEFAULT_FROM_EMAIL, 198

DEFAULT_LANGUAGE, 329

DEFAULT_MERGE_MESSAGE, 328

DEFAULT_MERGE_STYLE, 329

DEFAULT_PAGE_LIMIT, 344

DEFAULT_PULL_MESSAGE, 329

DEFAULT_RESTRICTED_COMPONENT, 327

DEFAULT_SHARED_TM, 329

DEFAULT_TRANSLATION_PROPAGATION, 329

ENABLE_AVATARS, 330

ENABLE_HOOKS, 330

ENABLE_HTTPS, 330

ENABLE_SHARING, 330

GET_HELP_URL, 330

GITEA_CREDENTIALS, 330

GITEA_TOKEN, 331

GITEA_USERNAME, 331

GITHUB_CREDENTIALS, 332

GITHUB_TOKEN, 332

GITHUB_USERNAME, 332

GITLAB_CREDENTIALS, 331

GITLAB_TOKEN, 332

GITLAB_USERNAME, 332

GOOGLE_ANALYTICS_ID, 333

HIDE_REPO_CREDENTIALS, 333

HIDE_VERSION, 333
INTERLEDGER_PAYMENT_POINTERS, 333
IP_BEHIND_REVERSE_PROXY, 333
IP_PROXY_HEADER, 334
IP_PROXY_OFFSET, 334
LEGAL_URL, 334
LICENSE_EXTRA, 335
LICENSE_FILTER, 335
LICENSE_REQUIRED, 335
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 336
LOCALIZE_CDN_PATH, 336
LOCALIZE_CDN_URL, 336
LOGIN_REQUIRED_URLS, 336
LOGIN_REQUIRED_URLS_EXCEPTIONS, 336
MACHINE_TRANSLATION_SERVICES, 337
MATOMO_SITE_ID, 337
MATOMO_URL, 337
MT_APERTIUM_API, 338
MT_AWS_ACCESS_KEY_ID, 338
MT_AWS_REGION, 338
MT_AWS_SECRET_ACCESS_KEY, 338
MT_BAIDU_ID, 339
MT_BAIDU_SECRET, 339
MT_DEEPL_API_URL, 339
MT_DEEPL_KEY, 339
MT_GOOGLE_CREDENTIALS, 340
MT_GOOGLE_KEY, 340
MT_GOOGLE_LOCATION, 340
MT_GOOGLE_PROJECT, 340
MT_LIBRETRANSLATE_API_URL, 340
MT_LIBRETRANSLATE_KEY, 340
MT_MICROSOFT_BASE_URL, 341
MT_MICROSOFT_COGNITIVE_KEY, 341
MT_MICROSOFT_ENDPOINT_URL, 341
MT_MICROSOFT_REGION, 341
MT_MODERNMT_KEY, 341
MT_MODERNMT_URL, 341
MT_MYMMEMORY_EMAIL, 342
MT_MYMMEMORY_KEY, 342
MT_MYMMEMORY_USER, 342
MT_NETEASE_KEY, 342
MT_NETEASE_SECRET, 342
MT_SAP_BASE_URL, 343
MT_SAP_PASSWORD, 343
MT_SAP_SANDBOX_APIKEY, 343
MT_SAP_USE_MT, 343
MT_SAP_USERNAME, 343
MT_SERVICES, 337
MT_TMSERVER, 342
MT_YANDEX_KEY, 342
MT_YOUDAO_ID, 343
MT_YOUDAO_SECRET, 343
NEARBY_MESSAGES, 344
PAGURE_CREDENTIALS, 344
PAGURE_TOKEN, 344
PAGURE_USERNAME, 344
PIWIK_SITE_ID, 337
PIWIK_URL, 337
PRIVACY_URL, 345
RATELIMIT_ATTEMPTS, 345
RATELIMIT_LOCKOUT, 345
RATELIMIT_WINDOW, 345
REGISTRATION_ALLOW_BACKENDS, 346
REGISTRATION_CAPTCHA, 346
REGISTRATION_EMAIL_MATCH, 346
REGISTRATION_OPEN, 346
REPOSITORY_ALERT_THRESHOLD, 347
REQUIRE_LOGIN, 347
SECRET_KEY, 198
SENTRY_DSN, 347
SERVER_EMAIL, 198
SESSION_COOKIE_AGE_AUTHENTICATED, 347
SESSION_ENGINE, 198
SIMPLIFY_LANGUAGES, 347
SINGLE_PROJECT, 348
SITE_DOMAIN, 348
SITE_TITLE, 348
SPECIAL_CHARS, 348
SSH_EXTRA_ARGS, 349
STATUS_URL, 349
SUGGESTION_CLEANUP_DAYS, 349
UPDATE_LANGUAGES, 349
URL_PREFIX, 349
VCS_BACKENDS, 350
VCS_CLONE_DEPTH, 350
WEBLATE_ADDONS, 350
WEBLATE_EXPORTERS, 351
WEBLATE_FORMATS, 351
WEBLATE_GPG_IDENTITY, 351
WEBSITE_REQUIRED, 352
setupgroups
 weblate admin command, 378
setupgroupsopción de línea de comando
 --no-privs-update, 378
 --no-projects-update, 378
setuplang
 weblate admin command, 379
setuplangopción de línea de comando
 --no-update, 379
show
 wlcopción de línea de comando, 144
SIMPLIFY_LANGUAGES
 setting, 347
SINGLE_PROJECT
 setting, 348
SITE_DOMAIN
 setting, 348
SITE_TITLE
 setting, 348
SPECIAL_CHARS
 setting, 348
SSH_EXTRA_ARGS
 setting, 349
statistics

wlcopción de línea de comando, 145
 STATUS_URL
 setting, 349
 store_post_load() (método de *weblate.addons.base.BaseAddon*), 414
 string resources
 file format, 76
 SUGGESTION_CLEANUP_DAYS
 setting, 349

T

translation
 bilingual, 68
 iPad, 77
 iPhone, 77
 monolingual, 68
 TS
 file format, 76

U

unit_pre_create() (método de *weblate.addons.base.BaseAddon*), 414
 unlock
 wlcopción de línea de comando, 145
 unlock_translation
 weblate admin command, 378
 UPDATE_LANGUAGES
 setting, 349
 updatechecks
 weblate admin command, 379
 updategit
 weblate admin command, 379
 upload
 wlcopción de línea de comando, 145
 URL_PREFIX
 setting, 349

V

variables de entorno
 CELERY_BACKUP_OPTIONS, 155, 172
 CELERY_BEAT_OPTIONS, 155, 172
 CELERY_MAIN_OPTIONS, 155, 172
 CELERY_MEMORY_OPTIONS, 155, 172
 CELERY_NOTIFY_OPTIONS, 155, 172
 CELERY_TRANSLATE_OPTIONS, 155, 172
 POSTGRES_ALTER_ROLE, 167
 POSTGRES_CONN_MAX_AGE, 167
 POSTGRES_DATABASE, 167
 POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 168
 POSTGRES_HOST, 167
 POSTGRES_PASSWORD, 167
 POSTGRES_PASSWORD_FILE, 167
 POSTGRES_PORT, 167
 POSTGRES_SSL_MODE, 167
 POSTGRES_USER, 167
 REDIS_DB, 168
 REDIS_HOST, 168

REDIS_PASSWORD, 168
 REDIS_PORT, 168
 REDIS_TLS, 168
 REDIS_VERIFY_SSL, 169
 ROLLBAR_ENVIRONMENT, 171
 ROLLBAR_KEY, 171
 SENTRY_DSN, 171
 SENTRY_ENVIRONMENT, 171
 SOCIAL_AUTH_SLACK_SECRET, 167
 WEB_WORKERS, 155, 172
 WEBLATE_ADD_ADDONS, 172
 WEBLATE_ADD_APPS, 171
 WEBLATE_ADD_AUTOFIX, 171
 WEBLATE_ADD_CHECK, 171
 WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 159
 WEBLATE_ADMIN_EMAIL, 155--157, 163
 WEBLATE_ADMIN_NAME, 155--157
 WEBLATE_ADMIN_PASSWORD, 151, 155--157
 WEBLATE_ADMIN_PASSWORD_FILE, 157
 WEBLATE_AKISMET_API_KEY, 160, 386
 WEBLATE_ALLOWED_HOSTS, 157, 198, 202, 348
 WEBLATE_API_RATELIMIT_ANON, 99, 161
 WEBLATE_API_RATELIMIT_USER, 99, 161
 WEBLATE_AUTH_LDAP_BIND_DN, 163
 WEBLATE_AUTH_LDAP_BIND_PASSWORD, 163
 WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS, 164
 WEBLATE_AUTH_LDAP_SERVER_URI, 163
 WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 163
 WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 163
 WEBLATE_AUTH_LDAP_USER_SEARCH, 164
 WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 164
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 164
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 164
 WEBLATE_AUTO_UPDATE, 170
 WEBLATE_BASIC_LANGUAGES, 161
 WEBLATE_BORG_EXTRA_ARGS, 162
 WEBLATE_CONTACT_FORM, 157
 WEBLATE_CSP_CONNECT_SRC, 160
 WEBLATE_CSP_FONT_SRC, 161
 WEBLATE_CSP_IMG_SRC, 160
 WEBLATE_CSP_SCRIPT_SRC, 160
 WEBLATE_CSP_STYLE_SRC, 160
 WEBLATE_DATABASE_BACKUP, 168
 WEBLATE_DEBUG, 156
 WEBLATE_DEFAULT_ACCESS_CONTROL, 160
 WEBLATE_DEFAULT_AUTO_WATCH, 161
 WEBLATE_DEFAULT_COMMITTER_EMAIL, 160
 WEBLATE_DEFAULT_COMMITTER_NAME, 160
 WEBLATE_DEFAULT_FROM_EMAIL, 157

- WEBLATE_DEFAULT_PULL_MESSAGE, 160
- WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 160
- WEBLATE_DEFAULT_SHARED_TM, 160
- WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 160
- WEBLATE_EMAIL_BACKEND, 170
- WEBLATE_EMAIL_HOST, 169
- WEBLATE_EMAIL_HOST_PASSWORD, 169
- WEBLATE_EMAIL_HOST_PASSWORD_FILE, 169
- WEBLATE_EMAIL_HOST_USER, 169
- WEBLATE_EMAIL_PORT, 169, 170
- WEBLATE_EMAIL_USE_SSL, 169, 170
- WEBLATE_EMAIL_USE_TLS, 169, 170
- WEBLATE_ENABLE_AVATARS, 161
- WEBLATE_ENABLE_HTTPS, 158, 231
- WEBLATE_GET_HELP_URL, 170
- WEBLATE_GITHUB_TOKEN, 159
- WEBLATE_GITHUB_USERNAME, 159
- WEBLATE_GITLAB_TOKEN, 159
- WEBLATE_GITLAB_USERNAME, 159
- WEBLATE_GOOGLE_ANALYTICS_ID, 159
- WEBLATE_GPG_IDENTITY, 160
- WEBLATE_HIDE_VERSION, 161
- WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 158
- WEBLATE_IP_PROXY_HEADER, 158
- WEBLATE_LEGAL_URL, 170
- WEBLATE_LICENSE_FILTER, 161
- WEBLATE_LICENSE_REQUIRED, 161
- WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LANGUAGE, 161
- WEBLATE_LOCALIZE_CDN_PATH, 171
- WEBLATE_LOCALIZE_CDN_URL, 171
- WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 159
- WEBLATE_LOGLEVEL, 156
- WEBLATE_LOGLEVEL_DATABASE, 156
- WEBLATE_MT_APERTIUM_APY, 162
- WEBLATE_MT_AWS_ACCESS_KEY_ID, 162
- WEBLATE_MT_AWS_REGION, 162
- WEBLATE_MT_AWS_SECRET_ACCESS_KEY, 162
- WEBLATE_MT_DEEPL_API_URL, 162
- WEBLATE_MT_DEEPL_KEY, 162
- WEBLATE_MT_GLOSBE_ENABLED, 163
- WEBLATE_MT_GOOGLE_CREDENTIALS, 162
- WEBLATE_MT_GOOGLE_KEY, 162
- WEBLATE_MT_GOOGLE_LOCATION, 162
- WEBLATE_MT_GOOGLE_PROJECT, 162
- WEBLATE_MT_LIBRETRANSLATE_API_URL, 162
- WEBLATE_MT_LIBRETRANSLATE_KEY, 162
- WEBLATE_MT_MICROSOFT_BASE_URL, 163
- WEBLATE_MT_MICROSOFT_COGNITIVE_KEY, 162
- WEBLATE_MT_MICROSOFT_ENDPOINT_URL, 162
- WEBLATE_MT_MICROSOFT_REGION, 162
- WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED, 163
- WEBLATE_MT_MODERNMT_KEY, 163
- WEBLATE_MT_MYMEMORY_ENABLED, 163
- WEBLATE_MT_SAP_BASE_URL, 163
- WEBLATE_MT_SAP_PASSWORD, 163
- WEBLATE_MT_SAP_SANDBOX_APIKEY, 163
- WEBLATE_MT_SAP_USE_MT, 163
- WEBLATE_MT_SAP_USERNAME, 163
- WEBLATE_NO_EMAIL_AUTH, 167
- WEBLATE_PAGURE_TOKEN, 160
- WEBLATE_PAGURE_USERNAME, 159
- WEBLATE_PRIVACY_URL, 170
- WEBLATE_RATELIMIT_ATTEMPTS, 161, 388
- WEBLATE_RATELIMIT_LOCKOUT, 161
- WEBLATE_RATELIMIT_WINDOW, 161
- WEBLATE_REGISTRATION_ALLOW_BACKENDS, 157
- WEBLATE_REGISTRATION_OPEN, 157
- WEBLATE_REMOVE_ADDONS, 172
- WEBLATE_REMOVE_APPS, 171
- WEBLATE_REMOVE_AUTOFIX, 171
- WEBLATE_REMOVE_CHECK, 171
- WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 159
- WEBLATE_REQUIRE_LOGIN, 159, 347
- WEBLATE_SAML_IDP_ENTITY_ID, 167
- WEBLATE_SAML_IDP_URL, 167
- WEBLATE_SAML_IDP_X509CERT, 167
- WEBLATE_SECURE_PROXY_SSL_HEADER, 158
- WEBLATE_SERVER_EMAIL, 157
- WEBLATE_SERVICE, 155, 172
- WEBLATE_SILENCED_SYSTEM_CHECKS, 160, 228
- WEBLATE_SIMPLIFY_LANGUAGES, 160
- WEBLATE_SITE_DOMAIN, 156, 200, 218, 348
- WEBLATE_SITE_TITLE, 156
- WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 166
- WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 166
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 166
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, 166
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID, 166
- WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 165
- WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY, 165
- WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET, 165
- WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 165

- 165
 - WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 165
 - WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 165
 - WEBLATE_SOCIAL_AUTH_FEDORA, 166
 - WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 165
 - WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY, 165
 - WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME, 165
 - WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET, 165
 - WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 165
 - WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID, 165
 - WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY, 165
 - WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET, 165
 - WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 166
 - WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 166
 - WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 166
 - WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 165
 - WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 165
 - WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 165
 - WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 165
 - WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 166
 - WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 166
 - WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 166
 - WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 166
 - WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 166
 - WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 166
 - WEBLATE_SOCIAL_AUTH_OPENSUSE, 166
 - WEBLATE_SOCIAL_AUTH_SLACK_KEY, 167
 - WEBLATE_SOCIAL_AUTH_UBUNTU, 166
 - WEBLATE_SSH_EXTRA_ARGS, 161
 - WEBLATE_STATUS_URL, 170
 - WEBLATE_TIME_ZONE, 157
 - WEBLATE_URL_PREFIX, 160
 - WEBLATE_WEBSITE_REQUIRED, 161
 - WEBLATE_WORKERS, 155, 172
 - WL_BRANCH, 318
 - WL_COMPONENT_NAME, 318
 - WL_COMPONENT_SLUG, 318
 - WL_COMPONENT_URL, 318
 - WL_ENGAGE_URL, 318
 - WL_FILE_FORMAT, 318
 - WL_FILEMASK, 318
 - WL_LANGUAGE, 318
 - WL_NEW_BASE, 318
 - WL_PATH, 318
 - WL_PREVIOUS_HEAD, 318
 - WL_PROJECT_NAME, 318
 - WL_PROJECT_SLUG, 318
 - WL_REPO, 318
 - WL_TEMPLATE, 318
 - WL_VCS, 318
 - VCS_BACKENDS
 - setting, 350
 - VCS_CLONE_DEPTH
 - setting, 350
 - version
 - wl opción de línea de comando, 144
- ## W
- WEB_WORKERS, 155, 172
 - Weblate (*clase en wlc*), 148
 - weblate admin command
 - add_suggestions, 369
 - auto_translate, 369
 - celery_queues, 370
 - checkgit, 370
 - cleanup_ssh_keys, 371
 - cleanuptrans, 370
 - commit_pending, 370
 - commitgit, 370
 - createadmin, 371
 - dump_memory, 371
 - dumpuserdata, 372
 - import_demo, 372
 - import_json, 372
 - import_memory, 373
 - import_project, 373
 - importuserdata, 375
 - importusers, 375
 - install_addon, 376
 - list_languages, 376
 - list_translators, 376
 - list_versions, 377
 - loadpo, 377
 - lock_translation, 377
 - move_language, 377
 - pushgit, 378
 - setupgroups, 378
 - setuplang, 379
 - unlock_translation, 378
 - updatechecks, 379
 - updategit, 379
 - WEBLATE_ADDONS
 - setting, 350
 - WEBLATE_ADMIN_EMAIL, 155--157, 163
 - WEBLATE_ADMIN_NAME, 155--157

WEBLATE_ADMIN_PASSWORD, 151, 155--157
 WEBLATE_ADMIN_PASSWORD_FILE, 157
 WEBLATE_AKISMET_API_KEY, 386
 WEBLATE_ALLOWED_HOSTS, 198, 202, 348
 WEBLATE_API_RATELIMIT_ANON, 99
 WEBLATE_API_RATELIMIT_USER, 99
 WEBLATE_EMAIL_HOST_PASSWORD, 169
 WEBLATE_EMAIL_PORT, 169, 170
 WEBLATE_EMAIL_USE_SSL, 169, 170
 WEBLATE_EMAIL_USE_TLS, 169, 170
 WEBLATE_ENABLE_HTTPS, 231
 WEBLATE_EXPORTERS
 setting, 351
 WEBLATE_FORMATS
 setting, 351
 WEBLATE_GPG_IDENTITY
 setting, 351
 WEBLATE_LOCALIZE_CDN_PATH, 171
 WEBLATE_RATELIMIT_ATTEMPTS, 388
 WEBLATE_REQUIRE_LOGIN, 347
 WEBLATE_SECURE_PROXY_SSL_HEADER, 158
 WEBLATE_SERVICE, 155
 WEBLATE_SILENCED_SYSTEM_CHECKS, 228
 WEBLATE_SITE_DOMAIN, 200, 218, 348
 WEBLATE_WORKERS, 155, 172
 WeblateConfig (*clase en wlc.config*), 148
 WeblateException, 148
 WEBSITE_REQUIRED
 setting, 352
 wlc, 142
 módulo, 148
 wlc.config
 módulo, 148
 wlc.main
 módulo, 149
 wlcopción de línea de comando
 --author-email, 146
 --author-name, 146
 --config, 144
 --config-section, 144
 --convert, 145
 --format, 144
 --fuzzy, 146
 --input, 146
 --key, 144
 --method, 146
 --output, 145
 --overwrite, 145
 --url, 144
 changes, 145
 cleanup, 145
 commit, 144
 download, 145
 list-components, 144
 list-languages, 144
 list-projects, 144
 list-translations, 144
 lock, 145

lock-status, 145
 ls, 144
 pull, 145
 push, 145
 repo, 145
 reset, 145
 show, 144
 statistics, 145
 unlock, 145
 upload, 145
 version, 144

X

XLIFF
 file format, 72
 XML
 file format, 84

Y

YAML
 file format, 83
 YAML Ain't Markup Language
 file format, 83