



The Weblate Manual

Versão 4.7.2

Michal Čihař

15 jul., 2021

1	Documentação de utilizador	1
1.1	Básico do Weblate	1
1.2	Registo e perfil de utilizador	1
1.3	Traduzir a usar o Weblate	10
1.4	Descarregar e enviar traduções	20
1.5	Glossário	24
1.6	Verificações e correções	26
1.7	Pesquisando	43
1.8	Fluxos de trabalho de tradução	49
1.9	Perguntas frequentes	52
1.10	Formatos de ficheiros suportados	60
1.11	Integração de controlo de versões	79
1.12	API REST do Weblate	86
1.13	Cliente Weblate	131
1.14	API Python do Weblate	135
2	Documentação de administrador	138
2.1	Instruções de configuração	138
2.2	Implantações de Weblate	199
2.3	Atualizando o Weblate	200
2.4	Fazer backup e mover o Weblate	206
2.5	Autenticação	212
2.6	Controlo de acesso	223
2.7	Projetos de tradução	232
2.8	Definições de idioma	249
2.9	Tradução contínua	252
2.10	Licenciar traduções	262
2.11	Processo de tradução	263
2.12	Verificações e correções	269
2.13	Tradução automática	277
2.14	Extensões	284
2.15	Memória de Tradução	299
2.16	Configuração	300
2.17	Amostra de configuração	329
2.18	Comandos de gerência	345
2.19	Anúncios	356
2.20	Lista de componentes	359
2.21	Módulos opcionais do Weblate	360
2.22	Personalizar o Weblate	365
2.23	Interface de gestão	367
2.24	Obter suporte para o Weblate	375

2.25	Documentos legais	378
3	Documentação de colaborador	380
3.1	Contribuir ao Weblate	380
3.2	Começa çar a contribuir código para o Weblate	382
3.3	Código-fonte do Weblate	386
3.4	Depuração do Weblate	387
3.5	Componentes internos do Weblate	388
3.6	Desenvolvendo extensões	390
3.7	Frontend do Weblate	392
3.8	Relatando problemas no Weblate	393
3.9	Conjunto de testes e integração contínua do Weblate	393
3.10	Esquemas de dados	395
3.11	Criando lançamento do Weblate	399
3.12	Segurança e privacidade	400
3.13	Sobre o Weblate	400
3.14	Licença	401
4	Histórico de alterações	402
4.1	Weblate 4.7.2	402
4.2	Weblate 4.7.1	402
4.3	Weblate 4.7	402
4.4	Weblate 4.6.2	403
4.5	Weblate 4.6.1	403
4.6	Weblate 4.6	404
4.7	Weblate 4.5.3	404
4.8	Weblate 4.5.2	404
4.9	Weblate 4.5.1	405
4.10	Weblate 4.5	405
4.11	Weblate 4.4.2	406
4.12	Weblate 4.4.1	406
4.13	Weblate 4.4	407
4.14	Weblate 4.3.2	407
4.15	Weblate 4.3.1	408
4.16	Weblate 4.3	408
4.17	Weblate 4.2.2	409
4.18	Weblate 4.2.1	409
4.19	Weblate 4.2	409
4.20	Weblate 4.1.1	410
4.21	Weblate 4.1	411
4.22	Weblate 4.0.4	412
4.23	Weblate 4.0.3	412
4.24	Weblate 4.0.2	412
4.25	Weblate 4.0.1	413
4.26	Weblate 4.0	413
4.27	Weblate 3.x series	414
4.28	Weblate 2.x series	425
4.29	Weblate 1.x series	436
4.30	Weblate 0.x series	440
	Índice de Módulos do Python	444
	HTTP Routing Table	445
	Índice	448

1.1 Básico do Weblate

1.1.1 Estrutura de projetos e componentes

No Weblate, as traduções são organizadas em projetos e componentes. Cada projeto pode conter vários componentes, os quais contêm traduções para idiomas individuais. O componente corresponde a um ficheiro traduzível (por exemplo, *GNU gettext* ou *Recurso de cadeias de Android*). Os projetos existem para ajudá-lo a organizar componentes em conjuntos lógicos (por exemplo, para agrupar todas as traduções usadas dentro de uma aplicação).

Internamente, cada projeto tem traduções para cadeias comuns propagadas em outros componentes dentro dele por predefinição. Isso alivia o fardo da tradução repetitiva e de várias versões. A propagação da tradução pode ser desativada por *Configuração de componente* a utilizar *Permitir propagação da tradução* caso as traduções devam divergir.

Veja também:

`../devel/integration`

1.2 Registo e perfil de utilizador

1.2.1 Registo

Todos podem procurar projetos, visualizar traduções ou sugerir traduções por predefinição. Somente utilizadores registados têm permissão para realmente gravar as alterações e são creditados para cada tradução feita.

Pode registar-se seguindo alguns passos simples:

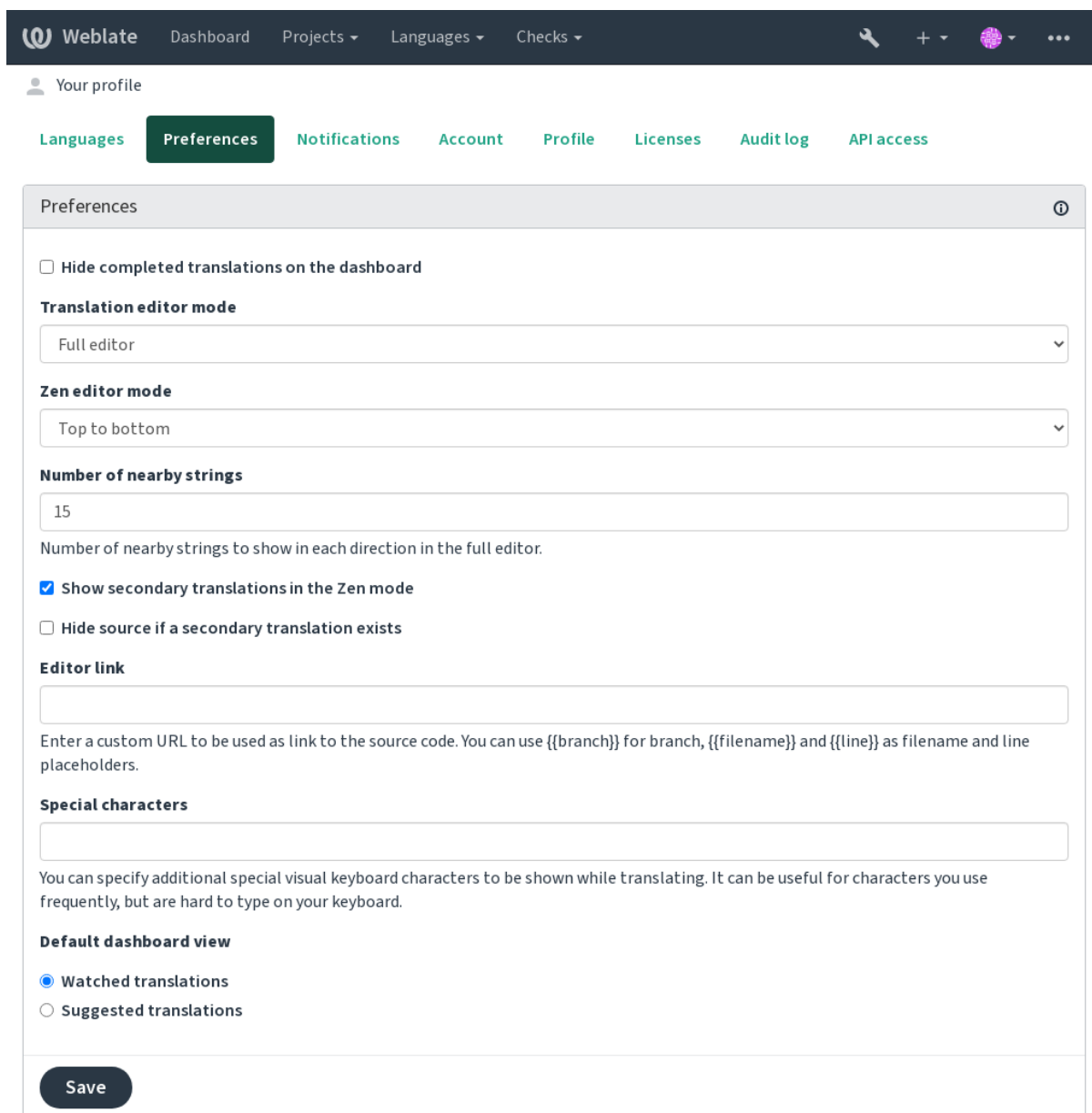
1. Preencha o formulário de registo com as suas credenciais.
2. Ative o registo seguindo a hiperligação no e-mail que receber.
3. Ajuste opcionalmente o seu perfil para escolher quais idiomas conhece.

1.2.2 Painei

Ao fazer login verá uma visão geral de projetos e componentes, bem como a respetiva progressão de tradução deles. Novo na versão 2.5.

Os componentes dos projetos que está a observar são mostrados por predefinição e cruzados com os idiomas da sua preferência.

Dica: Pode mudar para visualizações diferentes a usar as guias de navegação.



The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks' menus. On the right are icons for settings, a plus sign, a user profile, and a menu. Below the navigation bar is a 'Your profile' section. A row of tabs includes 'Languages', 'Preferences' (which is active and highlighted in dark green), 'Notifications', 'Account', 'Profile', 'Licenses', 'Audit log', and 'API access'. The 'Preferences' panel is open, showing various settings. It starts with a 'Hide completed translations on the dashboard' checkbox. Under 'Translation editor mode', there is a dropdown menu set to 'Full editor'. Under 'Zen editor mode', there is a dropdown menu set to 'Top to bottom'. The 'Number of nearby strings' is set to '15' with a text input field and a description: 'Number of nearby strings to show in each direction in the full editor.' There is a checked checkbox for 'Show secondary translations in the Zen mode' and an unchecked checkbox for 'Hide source if a secondary translation exists'. The 'Editor link' section has a text input field and 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.' The 'Special characters' section has a text input field and 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.' The 'Default dashboard view' section has two radio buttons: 'Watched translations' (which is selected) and 'Suggested translations'. At the bottom of the panel is a 'Save' button.

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

O menu tem estas opções:

- *Projetos > Visualizar todos os projetos* no menu principal mostra o estado da tradução, para cada projeto, na instância do Weblate.
- Selecionar um idioma no menu principal de *Idiomas* irá mostrar o estado da tradução de todos os projetos,

filtrado por um dos seus idiomas primários.

- *Traduções observadas* no Painel vai mostrar o estado da tradução apenas dos projetos que está observando, filtradas por os seus idiomas primários.

Além disso, o menu suspenso também pode mostrar qualquer quantidade de *listas de componentes*, conjuntos de componentes do projeto pré-configurados pelo administrador da Weblate, veja [Lista de componentes](#).

Pode configurar a sua exibição de painel predefinido pessoal na secção *Preferências* das configurações do perfil do utilizador.

Nota: Quando o Weblate estiver configurado para um único projeto a usar `SINGLE_PROJECT` no ficheiro `settings.py` (veja [Configuração](#)), o painel não será mostrado, pois o utilizador será redirecionado para um único projeto ou componente.

1.2.3 Perfil do utilizador

O perfil do utilizador é acessível clicando no ícone do utilizador no topo direito do menu superior e depois no menu *Configurações*.

O perfil do utilizador contém as suas preferências. Nome e endereço de e-mail são usados em commits de VCS, por isso mantenha essas informações precisas.

Nota: Todas as seleções de idiomas só oferecem idiomas traduzidos atualmente.

Dica: Solicite ou adicione outros idiomas que deseja traduzir clicando no botão para torná-los também disponíveis.


Idiomas

1.2.4 Idioma da interface





Escolha o idioma que deseja exibir na UI.


Idiomas traduzidos

Escolha quais idiomas prefere traduzir e eles serão oferecidos na página principal de projetos assistidos, para que tenha acesso mais fácil a todas essas traduções em cada um desses idiomas.

 Weblate

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


 Dashboard








































Watched translations 13

Suggested translations 5

Insights

Search



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

Powered by Weblate 4.7

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

Idiomas secundários

Pode definir quais idiomas secundários são lhe mostrados como um guia durante a tradução. Um exemplo pode ser visto na imagem a seguir, onde o idioma hebreu é mostrado como secundário:

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area displays a translation for the string 'Files' (English) into Hebrew ('קבצים') and Czech ('Soubory'). The interface includes a top navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. A sidebar on the right contains a 'Glossary' section and 'String information' details.

Translation Area:

- Hebrew:** קבצים
- English:** Files
- Czech:** Soubory
- Needs editing:** ☐
- Buttons:** Save, Suggest, Skip

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:** 12 seconds ago
- Source string age:** 13 seconds ago
- Translation file:** [weblate/locale/cs/LC_MESSAGES/django.po](#), string 1

Other languages: 3

Language	Translated string
Hebrew	קבצים
Hungarian	Fájlok
English	Files

1.2.5 Preferências

Visualização predefinida do painel

Na guia *Preferências*, pode escolher qual das visualizações disponíveis do painel de instrumentos deve-se apresentar por predefinição. Se escolher a lista de *Lista de componentes*, terá que selecionar qual lista de componentes será exibida a partir da *Lista de componentes predefinida* suspensa.

Veja também:

Lista de componentes

Hiperligação do editor

Uma ligação de código-fonte é mostrado no navegador web configurado no *Configuração de componente* por predefinição.

Dica: Ao definir o *Ligação do editor*, usa o editor local para abrir o ficheiro de código-fonte VCS de cadeias traduzidas. Pode usar *Template markup*.

Geralmente alguma coisa como `editor://open/?file={{filename}}&line={{line}}` é uma boa opção.

Veja também:

Pode encontrar mais informações sobre o registo de protocolos de URL personalizados para o editor na [documentação do Nette](#).

1.2.6 Notificações

Inscreva-se em várias notificações da guia *Notificações*. As notificações para eventos selecionados em projetos assistidos ou administrados serão lhe enviadas por e-mail.

Algumas das notificações são enviadas apenas para eventos nos seus idiomas (por exemplo, sobre novas cadeias para traduzir), enquanto algumas acionam no nível de componente (por exemplo, erros de fusão). Esses dois grupos de notificações são visualmente separados nas configurações.

Pode alternar notificações para projetos assistidos e projetos administrados e pode ser mais ajustado (ou silenciado) por projeto e componente. Visite a página de visão geral de componentes e selecione a escolha apropriada no menu *Observando*.

No caso de *Observar automaticamente projetos em contribuição* estar ativado, começará automaticamente a ver os projetos ao traduzir uma cadeia. O valor padrão depende de `DEFAULT_AUTO_WATCH`.



Nota: Não receberá notificações para as suas próprias ações.


Powered by [Weblate 4.7](#) [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.2.7 Conta

A guia *Conta* permite configurar detalhes básicos da conta, conectar vários serviços que pode usar para entrar no Weblate, remover a sua conta completamente ou descarregar os seus dados de utilizador (veja *Exportação de dados de utilizadores do Weblate*).

Nota: A lista de serviços depende da configuração do Weblate, mas pode ser feita para incluir sites populares como GitLab, GitHub, Google, Facebook ou Bitbucket ou outros provedores de OAuth 2.0.

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

 Your profile

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

Account

Username

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

Full name






E-mail

You can add another e-mail address below.


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

Save

Current user identities

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

Add new association

 **E-mail**

Removal

Account removal deletes all your private data.

Remove my account

User data

You can download all your private data.

Download user data

1.2.8 Perfil

Todos os campos desta página são opcionais e podem ser apagados a qualquer momento, e ao preenchê-los, dá-nos o seu consentimento para compartilhar esses dados onde quer que o seu perfil de utilizador apareça.

Um avatar pode ser mostrado para cada utilizador (dependendo de [ENABLE_AVATARS](#)). Estas imagens são obtidas utilizando <https://gravatar.com/>.

1.2.9 Licenças

1.2.10 Acesso API

Pode obter ou redefinir o seu token API de acesso aqui.

1.2.11 Registo de auditoria

O registo de auditoria rastreia as ações realizadas com a sua conta. Ele regista o endereço IP e o navegador para cada ação importante com a sua conta. As ações críticas também desencadeiam uma notificação a um endereço de e-mail principal.

Veja também:

Executar por trás de um proxy reverso

1.3 Traduzir a usar o Weblate

Obrigado pelo interesse em traduzir a usar o Weblate. Os projetos podem ser configurados para tradução direta ou por meio de sugestões feitas por utilizadores sem contas.

No geral, há dois modos de tradução:

- O projeto aceita traduções diretas
- O projeto aceita apenas sugestões, que são validadas automaticamente uma vez que uma quantidade definida de votos é alcançado

Por favor, veja [Fluxos de trabalho de tradução](#) para obter mais informações sobre fluxo de trabalho de tradução.

Opções para a visibilidade do projeto de tradução:

- Publicamente visível e todos podem contribuir
- Visível apenas para um certo grupo de tradutores

Veja também:

Controlo de acesso, Fluxos de trabalho de tradução

1.3.1 Projetos de tradução

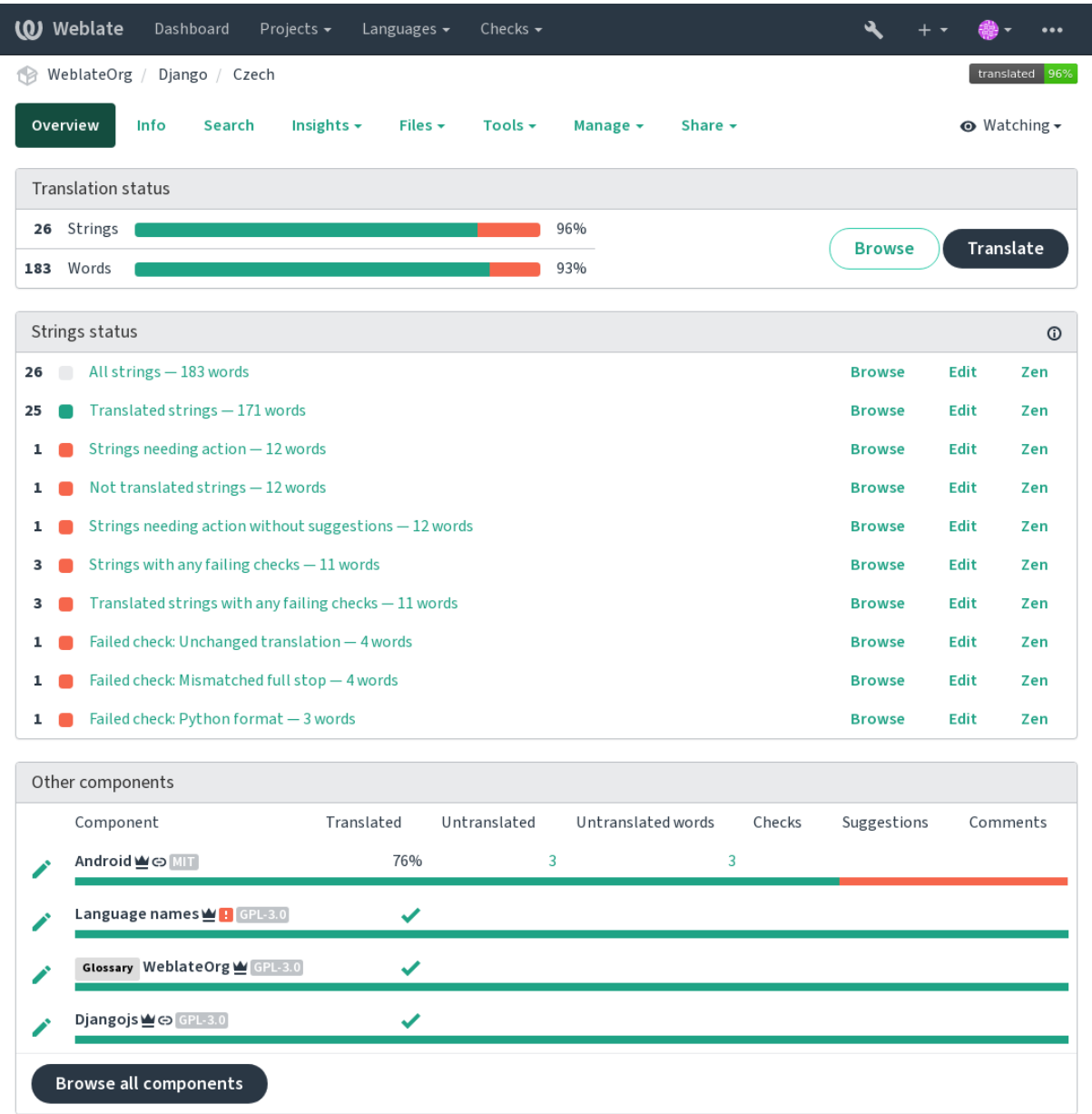
Os projetos de tradução possuem componentes; recursos para o mesmo software, livro ou projeto.

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

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.3.2 Ligações de tradução

Depois de navegar para um componente, um conjunto de ligações leva a sua tradução real. A tradução é ainda dividida em verificações individuais, como *Cadeias não traduzidas* ou *Cadeias que necessitam edição*. Se todo o projeto for traduzido, sem erro, *Todas as cadeias* ainda estão disponíveis. Alternativamente, pode usar o campo de pesquisa para encontrar uma cadeia ou termo específico.



Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.3.3 Sugestões

Nota: As permissões podem variar de acordo com a configuração da sua instância do Weblate.

Utilizadores anônimos só podem (por padrão) encaminhar sugestões. Isso ainda está disponível para os utilizadores autenticados, nos casos em que surge a incerteza sobre a tradução, a levar outros tradutores a revisá-la.

As sugestões são verificadas diariamente para remover as duplicatas e sugestões a corresponder à tradução atual.

1.3.4 Comentários

Três tipos de comentários podem ser publicados: para traduções, cadeias fonte ou para relatar erros de cadeias fonte quando esta funcionalidade é ativada a utilizar [Ativar revisões de fontes](#). Escolha o adequado para o tópico que deseja discutir. Os comentários de cadeia fonte são, em qualquer caso, bons para fornecer feedback sobre a cadeia original, por exemplo, que ele deve ser reformulado ou para fazer perguntas sobre ele.

Pode usar a sintaxe do Markdown em todos os comentários e mencionar outros utilizadores a usar @menção.

Veja também:

report-source, [Revisões de cadeias fonte](#), [Ativar revisões de fontes](#)

1.3.5 Variantes

As variantes são usadas para agrupar variantes de diferentes comprimentos da cadeia. O frontend do seu projeto pode, então, usar cadeias diferentes a depender do ecrã ou do tamanho da janela.

Veja também:

variants, [Variantes](#)

1.3.6 Etiquetas

As etiquetas são usadas para categorizar cadeias dentro de um projeto para personalizar ainda mais o fluxo de trabalho de localização (por exemplo, para definir categorias de cadeias).

Veja também:

labels

1.3.7 Traduzir

Na página de tradução, a cadeia fonte e uma área de edição para tradução dela são mostrados. Caso a tradução seja plural, são mostradas múltiplas cadeia fonte e áreas de edição, cada um descrito e rotulado em na quantidade de formas de plural que o idioma traduzido tem.

Todos os caracteres especiais de espaço em branco são sublinhados em vermelho e indicados com símbolos cinzentos. Mais de um espaço subsequente também é sublinhado em vermelho para alertar o tradutor para um possível problema de formatação.

Vários pedaços de informações extras podem ser mostrados nesta página, a maioria proveniente do código-fonte do projeto (como contexto, comentários ou onde a mensagem está a ser usada). Campos de tradução para quaisquer idiomas secundários, que os tradutores venham a selecionar nas preferências, serão mostrados (veja [Idiomas secundários](#)) acima da cadeia fonte.

Em baixo da tradução, tradutores vão encontrar sugestões feitas por outros, para serem aceites (✓), aceites com alterações (✎) ou apagadas (✖).

Plurais

Palavras a mudar a forma para levar em conta sua designação numérica são chamadas de plurais. Cada idioma tem uma própria definição de plurais. O inglês, por exemplo, tem suporte a uma. Na definição singular de, por exemplo, «car» (carro), implicitamente um carro é referenciado, enquanto na definição plural, «carros» significa dois ou mais carros (ou o conceito de carros como substantivo). Idiomas como, por exemplo, tcheco ou árabe têm mais plurais e também as regras dele para os plurais são diferentes.

O Weblate tem total suporte a cada uma dessas formas, em cada respectivo idioma (a traduzir cada plural separadamente). A quantidade de campos e como ele é usado na aplicação ou projeto traduzido depende da forma de plural configurada. Weblate mostra as informações básicas e [Language Plural Rules](#) do Unicode Consortium tem informações mais detalhadas.

Veja também:

[Fórmula de plural](#)

Weblate
 Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate
 translated 96%

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

Position and priority

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

%(count)s slovo

Czech, Few

%(count)s slova

Czech, Other

%(count)s slov

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

☐ Needs editing

Save Suggest Skip

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

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.h
tml:149

String age

7 seconds ago

Source string age

7 seconds ago

Translation file

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

Atalhos de teclado

Alterado na versão 2.18: Os atalhos do teclado foram renovados em 2.18 para reduzir a possibilidade de colidir com o atalhos predefinidos de navegadores ou sistemas.

Os seguintes atalhos de teclado podem ser utilizados durante a tradução:

Atalho de teclado	Descrição
Alt+Home	De momento, esta tradução está bloqueada.
Alt+End	De momento, esta tradução está bloqueada.
Alt+PageUp ou Ctrl ↑ ou Alt ↑ ou Cmd ↑	De momento, esta tradução está bloqueada.
Alt+PageDown ou Ctrl+↓ ou Alt+↓ ou Cmd+↓	De momento, esta tradução está bloqueada.
Alt+Enter ou Ctrl+Enter ou Cmd+Enter	Gravar tradução atual.
Ctrl+Shift+Enter ou Cmd+Shift+Enter	Desmarca a tradução como a precisar de edição e a envia.
Ctrl+E ou Cmd+E	Muda o foco ao editor de tradução.
Ctrl+U ou Cmd+U	Muda o foco ao editor de comentários.
Ctrl+M ou Cmd+M	Mostra a guia <i>Sugestões automáticas</i> , consulte <i>Sugestões automáticas</i> .
Ctrl+1 até Ctrl+9 ou Cmd+1 até Cmd+9	Copia objetos colocáveis de determinada quantidade da cadeia fonte.
Ctrl+M+1 até 9 ou Cmd+M+1 até 9	É passado como um parâmetro único que consiste o nome de uma tradução atual.
Ctrl+I+1 até 9 ou Cmd+I+1 até 9	Ignore um item na lista de verificações falhadas.
Ctrl+J ou Cmd+J	Mostra a guia de <i>Cadeias próximas</i> .
Ctrl+S ou Cmd+S	Muda o foco para o campo de pesquisa.
Ctrl+O ou Cmd+O	Copia a cadeia fonte.
Ctrl+Y ou Cmd+Y	Marca ou desmarca o marcador <i>Necessita edição</i> .

Teclado visual

Uma pequena linha de teclado visual é mostrada logo acima do campo de tradução. Isto pode ser útil para manter a pontuação local em mente (pois a linha é local para cada idioma) ou ter caracteres que seriam de difícil digitação.

Os símbolos mostrados são apresentados em três categorias:

- Caracteres configurados pelo utilizador definidos em *Perfil do utilizador*
- Caracteres por idioma fornecidos pelo Weblate (por exemplo, citações ou caracteres específicos RTL)
- Caracteres configurados a usar *SPECIAL_CHARS*

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, the breadcrumb 'WeblateOrg / Django / Hebrew / Translate' is visible. A progress bar indicates 'translated 92%'. The main area shows the string 'Files' being translated from English to Hebrew. The Hebrew text 'קבצים' is entered. There are buttons for 'Save', 'Suggest', and 'Skip'. Below the main area, there are tabs for 'Nearby strings', 'Comments', 'Automatic suggestions', and 'Other languages'. A table shows the string 'Files' in English and its translations in Czech ('Soubory'), Hungarian ('Fájlok'), and Hebrew ('קבצים'). The right sidebar contains panels for 'Glossary', 'String information', 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Contexto da tradução

Esta descrição contextual fornece informações relacionadas sobre a cadeia atual.

Atributos da cadeia Coisas como ID da mensagem, contexto (`msgctxt`) ou localização no código-fonte.

Capturas de ecrã Capturas de ecrã podem ser enviadas ao Weblate para melhor informar os tradutores sobre onde e como a cadeia é usada, veja [Contexto visual para cadeias](#).

Cadeias próximas Exibe mensagens próximas do ficheiro de tradução. Estas também são geralmente usadas num contexto semelhante e se mostram úteis para manter a tradução consistente.

Outras ocorrências No caso de uma mensagem aparecer em vários lugares (por exemplo, vários componentes), esta guia mostra todos eles se forem considerados inconsistentes (veja [Inconsistente](#)). Pode escolher qual usar.

Memória de tradução Veja cadeias semelhantes traduzidas no passado, veja [Memória de Tradução](#).

Glossário Exibe termos do glossário do projeto usados na mensagem atual.

Alterações recentes Lista de pessoas que modificaram esta mensagem recentemente a usar Weblate.

Projeto Informações do projeto, como instruções para tradutores ou um diretório, ou ligação para a cadeia no repositório do sistema de controlo de versão que o projeto usa.

Se quiser ligações diretos, o formato da tradução tem que ter suporte.

Histórico de tradução

Cada alteração é por predefinição (a menos que desativada nas configurações dos componentes) gravada no banco de dados e pode ser revertida. Opcionalmente, ainda se pode reverter qualquer coisa no sistema de controle de versão subjacente.

Comprimento da cadeia traduzida

Weblate pode limitar o comprimento de uma tradução em várias formas para garantir a cadeia traduzida não é muito longa:

- A limitação padrão para tradução é dez vezes maior do que a cadeia fonte. Isso pode ser desativado em `LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH`. Caso esteja a atingir isso, ele também pode ser causado por uma tradução monolíngue ser erroneamente configurada como bilingue, a fazer o Weblate errar a chave de tradução para o texto fonte em vez da cadeia fonte. Veja [Formatos bilíngues e monolíngues](#) para obter mais informações.
- Comprimento máximo em caracteres definidos por ficheiro de tradução ou um sinalizador, consulte [Tamanho máximo da tradução](#).
- Tamanho máximo renderizado em pixels definido por sinalizadores, veja [Tamanho máximo da tradução](#).

1.3.8 Sugestões automáticas

Baseado na configuração e o seu idioma traduzido, o Weblate fornece sugestões de várias ferramentas de tradução e memória-tradução. Todas as traduções de máquina estão disponíveis numa única guia de cada página de tradução.

Veja também:

Encontra a lista de ferramentas suportadas em [Tradução automática](#).

1.3.9 Tradução automática

Pode usar a tradução automática para a iniciar a tradução com base em fontes externas. Esta ferramenta se chama *Tradução automática*, acessível no menu *Ferramentas*, uma vez que tenha selecionado um componente e um idioma:

Automatic translation

Automatic translation takes existing translations in this project and pushes them to a different branch, to fix inconsistent translations or to transfer them to a new component. It can be used to push translations to a different branch, to fix inconsistent translations or to transfer them to a new component. It can be used to push translations to a different branch, to fix inconsistent translations or to transfer them to a new component.

Automatic translation via machine translation uses active machine translation services to get the best possible translations and applies them in this project.

Automatic translation mode

Add as suggestion

Search filter

Strings needing action

Automatic translation source

☐ Other translation components ☒ Machine translation

Machine translation engines

Search...

Available:	Chosen:
Weblate	Weblate
Weblate Translation Memory	

Score threshold

80

Apply

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Dois modos de operação são possíveis:

- Usar outros componentes do Weblate como fonte para traduções.
- Usar serviços selecionados de tradução automática com traduções acima de um certo limite de qualidade.

Também pode escolher quais cadeias devem ser traduzidas automaticamente.

Aviso: Tenha em mente que isso substituirá as traduções existentes se empregadas com filtros amplos, como *Todos as cadeias*.

Útil em várias situações, como a consolidação da tradução entre diferentes componentes (por exemplo, a aplicação e o site dela) ou quando estiver a iniciar uma tradução para um novo componente a usar traduções existentes (memória de tradução).

Veja também:

Manter traduções iguais entre componentes

1.3.10 Limitação de taxa

Para evitar abusos na interface, uma limitação de taxa é aplicada a várias operações como pesquisa, envio de formulários de contato ou tradução. Se afetado por isto, fica bloqueado por um certo período até que possa executar a operação novamente.

Os limites predefinidos e o ajuste fino estão descritos no manual administrativo, veja [Limitação de taxa](#).

1.3.11 Procurar e substituir

Altere a terminologia efetivamente ou execute a correção em massa das cadeias a usar *Pesquisar e substituir* no menu *Ferramentas*.

Dica: Não se preocupe em estragar as cadeias. Este é um processo de dois passos que mostra uma pré-visualização das cadeias editados antes que a alteração real seja confirmada.

1.3.12 Edição em massa

A edição em massa permite realizar uma operação sobre a quantidade de cadeias. Define os textos a procurar por eles e configura algo a ser feito para corresponder-lhes. As seguintes operações são suportadas:

- Alterar o estado da cadeia (por exemplo, para aprovar todas as cadeias não revistas).
- Ajustar os sinalizadores de tradução (veja [Personalizando comportamento a usar marcadores](#))
- Ajustar as etiquetas de cadeias (veja labels)

Dica: Esta ferramenta é chamada *Editor em massa*, acessível no menu *Ferramentas* de cada projeto, componente ou tradução.

Veja também:

[Extensão de edição em massa](#)

1.4 Descarregar e enviar traduções

Pode exportar ficheiros de uma tradução, fazer alterações e importá-los novamente. Isso permite trabalhar off-line e depois mesclar mudanças de volta na tradução existente. Isso funciona mesmo que tenha sido alterado entretanto.

Nota: As opções disponíveis podem ser limitadas por configurações de [controle de acesso](#).

1.4.1 Descarregar traduções

A partir do painel do projeto ou componente, ficheiros traduzíveis podem ser descarregados no menu *Ficheiros*.

A primeira opção é descarregar o ficheiro no formato original, uma vez que ele é armazenado no repositório. Neste caso, quaisquer alterações pendentes na tradução estão a ser comprometidas e o ficheiro atualizado é produzido sem nenhuma conversão.

Pode também descarregar a tradução convertida num dos formatos de localização amplamente utilizados. Os ficheiros convertidos serão enriquecidos com dados fornecidos no Weblate, como contexto adicional, comentários ou marcadores. Vários formatos de ficheiro estão disponíveis através do menu *Ficheiros* ↓ *Personalizar a descarrega*:

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

Dica: The content available in the converted files differs based on file format features, you can find overview in *Capacidades dos tipos de tradução*.

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The 'Files' dropdown menu is open, showing options: 'Download translation', 'Customize download' (highlighted), and 'Upload translation'. Below the menu, there is a table of quick downloads. The 'Customize download' section is also visible, showing a dropdown for 'All strings' and a 'File format' section with radio buttons for various formats: gettext PO (selected), XLIFF with gettext extensions, XLIFF 1.1, TBX, TMX, gettext MO, CSV, XLSX, JSON, Android String Resource, and iOS strings. A 'Download' button is at the bottom of the customize section.

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

Customize download

All strings ▾

File format

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

Download

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Veja também:

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

1.4.2 Enviar traduções

Quando tiver feito as suas alterações, use *Enviar tradução* no menu *Ficheiros*.

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb trail shows 'WeblateOrg / Django / Czech' with a 'translated 96%' indicator. A secondary navigation bar includes 'Overview', 'Info', 'Search', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. The 'Files' menu is open, showing options: 'Download translation', 'Customize download', and 'Upload translation'. The 'Upload translation' option is highlighted. Below the menu, the 'Upload' dialog is displayed. It contains a message: 'The uploaded file will be merged with the current translation. To overwrite already translated strings, don't forget to turn it on.' The dialog has several sections: 'File' with a 'Choose File' button and 'No file chosen' text; 'File upload mode' with radio buttons for 'Add as translation', 'Add as suggestion', 'Add as translation needing edit', and 'Replace existing translation file'; 'Processing of strings needing edit' with a dropdown menu set to 'Do not import'; 'Conflict handling' with a dropdown menu set to 'Update translated strings' and a note 'Whether to overwrite existing translations if the string is already translated.'; 'Author name' with a text input field containing 'Weblate Test'; and 'Author e-mail' with a text input field containing 'weblate@example.org'. At the bottom of the dialog is an 'Upload' button.

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Formatos de ficheiros suportados

Todos ficheiros num formato de ficheiro suportado pode ser enviado, mas ainda é recomendado usar o mesmo formato de ficheiro como o para a tradução, caso contrário, alguns recursos podem não ser traduzidos corretamente.

Veja também:

Formatos de ficheiros suportados

O ficheiro enviado é mesclado para atualizar a tradução, substituindo as entradas existentes por predefinição (pode ser desativado ou ativado na caixa de diálogo de envio).

Métodos de importação

Estas são as opções apresentadas ao enviar ficheiros de tradução:

Adicionar como tradução (`translate`) Traduções importadas são adicionadas como traduções. Este é o caso de uso mais comum e o comportamento predefinido.

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

Adicionar como sugestão (`suggest`) As traduções importadas são adicionadas como sugestões, faça isso quando quiser ter as suas cadeias enviadas serem revisadas.

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

Adicionar como tradução que necessita de edição (`fuzzy`) As traduções importadas são adicionadas como traduções que necessitam de edição. Isso pode ser útil quando quer que as traduções sejam usadas, mas também revisadas.

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

Substituir ficheiro de tradução existente (`replace`) O ficheiro existente é substituído por novo conteúdo. Isso pode levar à perda de traduções existentes, use com cuidado.

Atualizar cadeias fonte (`source`) Atualiza cadeias fonte em ficheiro de tradução bilíngue. Isso é semelhante ao que *Atualizar ficheiros PO para coincidir com POT (msgmerge)* faz.

Esta opção é suportada apenas para alguns formatos de ficheiro.

Adicionar cadeias (`add`) Adiciona cadeias à tradução. Pula a que já existe.

Caso queira adicionar cadeias e atualizar traduções existentes, envie do ficheiro pela segunda vez com *Adicionar como tradução*.

Esta opção está disponível apenas com *Gerir cadeias* ativado.

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

Veja também:

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

Gestão de conflitos

Define como lidar com cadeias enviadas que já são traduzidas.

Cadeias necessitando de edição

Há também uma opção de como lidar com cadeias que necessitam de edição no ficheiro importado. Tais cadeias podem ser manuseados de uma das três maneiras seguintes: «Não importar», «Importar como cadeia que necessita edição» ou «Importar como traduzido».

Substituindo autoria

Com permissões administrativas, também pode especificar a autoria do ficheiro enviado. Isso pode ser útil no caso de ter recebido o ficheiro de outra maneira e quiser mesclá-lo em traduções existentes enquanto credita corretamente o autor real.

1.5 Glossário

Cada projeto pode incluir um ou mais glossários como um atalho para armazenar terminologia. O glossário facilita a manutenção da consistência da tradução.

Um glossário para cada idioma pode ser gerido sozinho, mas eles são armazenados juntos como um único componente que ajuda os administradores de projeto e tradutores multilingues a manter alguma consistência entre os idiomas também. Os termos do glossário que contêm palavras da cadeia atualmente traduzida são exibidos na barra lateral do editor de tradução.

1.5.1 Gestão de glossários

Alterado na versão 4.5: Glossários são agora componentes de tradução comuns e pode usar todos os recursos da Weblate neles — a comentar, a armazenar num repositório remoto ou a adicionar explicações.

Use qualquer componente como um glossário a ligar *Usar como um glossário*. Pode criar vários glossários para um projeto.

Um glossário vazio para um determinado projeto é criado automaticamente com o projeto. Os glossários são compartilhados entre todos os componentes de um mesmo projeto e opcionalmente com outros projetos a utilizar *Compartilhar em projetos* do respectivo componente do glossário.

O componente do glossário se parece com qualquer outro componente da Weblate com etiqueta colorida adicionada:

Translation status

2	Strings	100%
3	Words	100%

Buttons: Add new glossary term, Browse, Translate

Strings status

2	All strings — 3 words	Browse	Edit	Zen
2	Translated strings — 3 words	Browse	Edit	Zen

Other components

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

Buttons: Browse all components

Powered by Weblate 4.7 About Weblate Legal Contact Documentation Donate to Weblate

Pode navegar por todos os termos do glossário:

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 path is 'WeblateOrg / Glossary WeblateOrg / Czech / Browse'. A status bar indicates 'translated 100%'. The main area shows a table with two columns: English and Czech. The first row contains 'machine translation' and 'strojový překlad'. The second row contains 'project' and 'projekt'. Navigation controls at the top include 'All strings', 'Source string', and 'Add new glossary term'. At the bottom, a footer mentions 'Powered by Weblate 4.7' and provides links for 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

ou edite-os como qualquer tradução.

1.5.2 Termos de glossário

Os termos de glossário são traduzidos da mesma forma que as cadeias comuns. Pode alternar recursos adicionais a utilizar o menu *Ferramenta* para cada termo.

This screenshot provides a detailed view of the Weblate Glossary interface. The main panel shows a 'Glossary term' for 'project' in English and 'projekt' in Czech. It includes a 'Needs editing' checkbox, an 'Explanation' field, and buttons for 'Save', 'Suggest', 'Skip', and 'Tools'. The 'Tools' menu is open, showing options like 'Delete string', 'Mark as read-only', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. On the right, a 'Glossary' sidebar shows the current term and a '+ Add term to glossary' button. Below it, a 'String information' panel shows 'String age' and 'Source string age'. At the bottom, a 'Nearby strings' section shows a list of related terms. The footer is identical to the previous screenshot.

Termos não traduzíveis

Novo na versão 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.

Veja também:

Personalizando comportamento a usar marcadores

Traduções proibidas

Novo na versão 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.

Veja também:

Personalizando comportamento a usar marcadores

Terminologia

Novo na versão 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.

Veja também:

Personalizando comportamento a usar marcadores

Variantes

As variantes são uma forma genérica de agrupar cadeias. Todas as variantes do termo são listadas no glossário lateral ao traduzir.

Dica: Pode usar isso para adicionar abreviações ou expressões mais curtas para um termo.

Veja também:

variants

1.6 Verificações e correções

As verificações de qualidade ajudam a apanhar erros comuns do tradutor, garantindo que a tradução esteja em boa forma. As verificações podem ser ignoradas em caso de falsos positivos.

Quando enviar uma tradução com uma verificação a falhar será imediatamente mostrada ao utilizador:

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

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

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

1/1

Custom search

'%(count)s word'

Position

Zen

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

Czech, Few

několik slov

Czech, Other

%(count)s slov

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

Needs editing

Save

Suggest

Skip

Things to check

Python format

Following format strings are missing: %(count)s

Dismiss

For all languages

Missing plurals

Some plural forms are not translated

Dismiss

For all languages

Glossary

English

Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

11 seconds ago

Source string age

11 seconds ago

Translation file

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

pending

Nearby strings

Comments

Automatic suggestions

Other languages

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.6. Verificações e correções

27

1.6.1 Correções automáticas

Além de *Verificações de qualidade*, o Weblate pode corrigir automaticamente alguns erros comuns em cadeias traduzidas. Use-o com cuidado para não causar erros por meio disto.

Veja também:

AUTOFIX_LIST

1.6.2 Verificações de qualidade

O Weblate emprega uma ampla gama de verificações de qualidade em cadeias. A secção a seguir descreve todos eles em mais detalhe. Há também verificações específicas de idiomas. Por favor, preencha um relatório de erro se alguma verificação for relatada por engano.

Veja também:

CHECK_LIST, *Personalizando comportamento a usar marcadores*

1.6.3 Verificações de tradução

Executado a cada alteração da tradução, ajuda os tradutores a manter traduções de boa qualidade.

Markup BBcode

BBcode na tradução não corresponde à fonte

BBCode representa marcação simples, como, por exemplo, destacar partes importantes de uma mensagem em fonte em negrito ou itálico.

Esta verificação garante que eles também estejam na tradução.

Nota: O método para detetar BBcode é atualmente bastante simples, então esta verificação pode produzir falsos positivos.

Palavras consecutivas duplicadas

O texto contém a mesma palavra duas vezes de seguida:

Novo na versão 4.1.

Verifica se não há palavras duplicadas consecutivas numa tradução. Isso geralmente indica um erro na tradução.

Dica: Esta verificação inclui regras específicas do idioma para evitar falsos positivos. Caso seja falso no seu caso, avise-nos. Veja *Relatando problemas no Weblate*.

Não segue o glossário

Novo na versão 4.5.

A tradução não segue os termos definidos num glossário.

Esta verificação tem de ser ligada a usar o sinalizador `check-glossary` (veja [Personalizando comportamento a usar marcadores](#)). Por favor, considere seguir antes de ativá-lo:

- Ele faz a correspondência exata das cadeias, espera-se que o glossário contenha termos em todas as variantes.
- A verificação de cada cadeia em relação ao glossário é dispendiosa, ela irá retardar qualquer operação na Weblate que envolva a execução de verificações como a importação de textos ou a tradução.

Veja também:

[Glossário](#), [Personalizando comportamento a usar marcadores](#), [Marcadores de tradução](#)

Espaço duplo

A tradução contém espaços duplos

Verifica se espaços duplos estão presentes na tradução para evitar falsos positivos em outras verificações relacionadas ao espaço.

A verificação é falsa quando espaços duplos são encontrados na fonte, o que significa que os espaços duplos são intencionais.

Cadeias formatadas

Verifica se a formatação em cadeias é replicada entre a fonte e a tradução. Omitir cadeias de formato na tradução geralmente causa problemas graves, de modo que a formatação em cadeias geralmente deve coincidir com a fonte.

O Weblate tem suporte a verificar cadeias de formato em vários idiomas. A verificação não é ativada automaticamente, somente se uma cadeia for sinalizada adequadamente (por exemplo, «c-format” para formato C). O Gettext adiciona-o automaticamente, mas provavelmente terá que adicioná-lo manualmente para outros formatos de ficheiro ou se os seus ficheiros PO não forem gerados por **xgettext**.

Isso pode ser feito por unidade (ver [Informações adicionais sobre cadeias fonte](#)) na [Configuração de componente](#). Tê-lo definido por componente é mais simples, mas pode levar a falsos positivos no caso de a cadeia não ser interpretada como uma cadeia de formatação, mas a sintaxe de textos de formato passa a ser usada.

Dica: Caso a verificação de formato específico não esteja disponível no Weblate, pode usar [Espaços reservados](#) genéricos.

Além de verificar, isso também destacará as cadeias de formatação para inseri-los facilmente em cadeias traduzidas:

Formato C

A cadeia de formato C não corresponde à fonte

Cadeia de formato simples	Há %d maçãs
Cadeia de formato de posição	O seu saldo é %1\$d %2\$s
Sinalize para ativar	<i>c-format</i>

Veja também:

Cadeias formatadas, Cadeias de formatação C, formatação de printf C

Formato C#

A cadeia de formato C# não corresponde à fonte

Cadeia de formato de posição	Há {0} maçãs
Sinalize para ativar	<i>c-sharp-format</i>

Veja também:

Cadeias formatadas, C# String Format

Literais de modelo de ECMAScript

Os literais de modelo de ECMAScript não correspondem à fonte

Interpolação	Há \${number} maçãs
Sinalize para ativar	<i>es-format</i>

Veja também:

Cadeias formatadas, Template literals

Interpolação de i18next

A interpolação de i18next não corresponde à fonte

Novo na versão 4.0.

Interpolação	Há {{number}} maçãs
Aninhamento	Há \$t(number) maçãs
Sinalize para ativar	<i>i18next-interpolation</i>

Veja também:

Cadeias formatadas, i18next interpolation

Formato Java

A cadeia de formato Java não corresponde à fonte

Cadeia de formato simples	Há %d maçãs
Cadeia de formato de posição	O seu saldo é %1\$d %2\$s
Sinalize para ativar	<i>java-format</i>

Veja também:

Cadeias formatadas, [Java Format Strings](#)

Formato de Mensagem Java

A cadeia de MessageFormat de Java não corresponde à fonte

Cadeia de formato de posição	Há {0} maçãs
Sinalize para ativar	<i>java-messageformat</i> ativa a verificação incondicionalmente
	<i>auto-java-messageformat</i> ativa a verificação somente se houver uma cadeia de formato na fonte

Veja também:

Cadeias formatadas, [Java MessageFormat](#)

Formato JavaScript

A cadeia de formato JavaScript não corresponde à fonte

Cadeia de formato simples	Há %d maçãs
Sinalize para ativar	<i>javascript-format</i>

Veja também:

Cadeias formatadas, [JavaScript formatting strings](#)

Formato Lua

O formato de cadeia Lua não corresponde com a fonte

Cadeia de formato simples	Há %d maçãs
Sinalize para ativar	<i>lua-format</i>

Veja também:

Cadeias formatadas, [Lua formatting strings](#)

Formato Object Pascal

Object Pascal format string does not match source

Cadeia de formato simples	Há %d maçãs
Sinalize para ativar	<i>object-pascal-format</i>

Veja também:

Cadeias formatadas, [Object Pascal formatting strings](#) [Free Pascal formatting strings](#) [Delphi formatting strings](#)

Espaços reservados de percentagem

Os espaços reservados de percentagem não correspondem à fonte

Novo na versão 4.0.

Cadeia de formato simples	Há %number% maçãs
Sinalize para ativar	<i>percent-placeholders</i>

Veja também:

Cadeias formatadas,

Formato Perl

A cadeia de formato Perl não corresponde à fonte

Cadeia de formato simples	Há %d maçãs
Cadeia de formato de posição	O seu saldo é %1\$d %2\$s
Sinalize para ativar	<i>perl-format</i>

Veja também:

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

Formato PHP

A cadeia de formato PHP não corresponde à fonte

Cadeia de formato simples	Há %d maçãs
Cadeia de formato de posição	O seu saldo é %1\$d %2\$s
Sinalize para ativar	<i>php-format</i>

Veja também:

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

Formato de chaveta Python

A cadeia de formato de chaves Python não corresponde à fonte

Cadeia de formato simples	Há {} maçãs
Cadeia de formato nomeado	O seu saldo é {amount} {currency}
Sinalize para ativar	<i>python-brace-format</i>

Veja também:

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

Formato Python

A cadeia de formato Python não corresponde à fonte

Cadeia de formato simples	Há %d maçãs
Cadeia de formato nomeado	O seu saldo é %(amount) %(currency)
Sinalize para ativar	<i>python-format</i>

Veja também:

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

Formato Qt

A cadeia de formato Qt não corresponde à fonte

Cadeia de formato de posição	Há %1 maçãs
Sinalize para ativar	<i>qt-format</i>

Veja também:

[Cadeias formatadas](#), [Qt QString::arg\(\)](#)

Forma plural Qt

A cadeia de formato de plural do Qt não corresponde à fonte

Cadeia de formato de plural	Há %Ln maçã(s)
Sinalize para ativar	<i>qt-plural-format</i>

Veja também:

[Cadeias formatadas](#), [Qt i18n guide](#)

Formato Ruby

A cadeia de formato Ruby não corresponde à fonte

Cadeia de formato simples	Há %d maçãs
Cadeia de formato de posição	O seu saldo é %1\$f %2\$s
Cadeia de formato nomeado	O seu saldo é %+ .2<amount>f %<currency>s
Cadeia de modelo nomeado	O seu saldo é %{amount} %{currency}
Sinalize para ativar	<i>ruby-format</i>

Veja também:

Cadeias formatadas, [Ruby Kernel#sprintf](#)

Formato Scheme

O formato de cadeia Scheme não corresponde com a fonte

Cadeia de formato simples	Há ~d maçãs
Sinalize para ativar	<i>scheme-format</i>

Veja também:

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

Formatação vue I18n

A formatação Vue I18n não corresponde com a fonte

Formatação nomeada	Há {count} maçãs
Formatação i18n de Rails	Há %{count} maçãs
Mensagens de localidade vinculadas	@:message.dio @:message.the_world!
Sinalize para ativar	<i>vue-format</i>

Veja também:

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

Foi traduzido

Esta cadeia foi traduzida no passado

Significa que uma cadeia já foi traduzida. Isso pode acontecer quando as traduções foram revertidas no VCS ou perdidas de outra forma.

Inconsistente

Esta cadeia tem mais que uma tradução neste projeto ou não é traduzida em alguns componentes.

O Weblate verifica traduções da mesma cadeia em todas as traduções de um projeto para ajudar a manter traduções consistentes.

A verificação falha em traduções diferentes de uma cadeia dentro de um projeto. Isso também pode levar a inconsistências nas verificações exibidas. Pode encontrar outras traduções desta cadeia na guia *Outras ocorrências*.

Nota: Esta verificação também é disparada no caso de a cadeia estar traduzida num componente e não em outro. Ela pode ser usada como uma maneira rápida de manusear manualmente cadeias que não estão traduzidas em alguns componentes apenas clicando no botão *Usar esta tradução* exibido em cada linha na guia *Outras ocorrências*.

Pode usar *Tradução automática* para automatizar a tradução de cadeias recém-adicionadas que já são traduzidas em outro componente.

Veja também:

Manter traduções iguais entre componentes

Letra Kashida utilizada

As letras kashida decorativas não devem ser usadas

Novo na versão 3.5.

As letras Kashida decorativas não devem ser usadas na tradução. Estas também são conhecidas como Tatweel.

Veja também:

[Kashida na Wikipédia](#)

Hiperligações de marcação

As ligações de Markdown não correspondem com a fonte

Novo na versão 3.5.

As ligações de Markdown não correspondem com a fonte.

Veja também:

Ligações Markdown

Referências de Markdown

As referências da ligação Markdown não correspondem à fonte

Novo na versão 3.5.

As referências de ligação Markdown não correspondem à fonte.

Veja também:

[Ligações Markdown](#)

Sintaxe de Markdown

A sintaxe de Markdown não corresponde à fonte

Novo na versão 3.5.

A sintaxe de Markdown não coincide com a fonte

Veja também:

[Elementos span Markdown](#)

Tamanho máximo da tradução

A tradução não deve exceder o comprimento dado

Verifica se as traduções são de comprimento aceitável para caber no espaço disponível. Isto apenas verifica o comprimento dos caracteres de tradução.

Ao contrário das outras verificações, o sinalizador deve ser definido como um par «chave:valor», como `max-length:100`.

Dica: Esta verificação analisa a quantidade de caracteres, o que pode não ser a melhor métrica ao utilizar fontes proporcionais para renderizar o texto. A verificação *Tamanho máximo da tradução* verifica a renderização real do texto.

O sinalizador `replacements:` também pode ser útil para expandir os objetos colocáveis antes de verificar a cadeia.

Tamanho máximo da tradução

O texto traduzido não deve exceder o tamanho determinado

Novo na versão 3.7.

O texto traduzido não deve exceder o tamanho determinado. Ele renderiza o texto com quebra de linhas e verifica se ele se encaixa nos limites determinados.

Esta verificação necessita de um ou dois parâmetros - largura máxima e quantidade máxima de linhas. Caso a quantidade de linhas não seja fornecida, um texto de uma linha é considerado .

Também pode configurar a fonte utilizada pelas diretivas `font-*` (veja *Personalizando comportamento a usar marcadores*), por exemplo, após os marcadores de tradução dizer que o texto renderizado com fonte ubuntu tamanho 22 deve caber em duas linhas e 500 pixels:

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

Dica: Queria definir as diretivas `font-*` na *Configuração de componente* para ter a mesma fonte configurada para todas as cadeias dentro de um componente. Pode sobrescrever esses valores por cadeia, caso precise personalizá-la por cadeia.

O sinalizador `replacements:` também pode ser útil para expandir os objetos colocáveis antes de verificar a cadeia.

Veja também:

Gerir letras, Personalizando comportamento a usar marcadores, Tamanho máximo da tradução

\n não correspondente

Quantidade de \n na tradução não corresponde à da fonte

Normalmente as novas linhas escapadas são importantes para a formatação da saída do programa. A verificação falha se a quantidade de literais \n na tradução não corresponder à fonte.

Dois pontos não correspondentes

Fonte e tradução não terminam com dois pontos

Verifica se os dois pontos são replicados entre a fonte e a tradução. A presença de dois pontos também é verificada para vários idiomas onde eles não pertencem (chinês ou japonês).

Veja também:

[Dois pontos na Wikipédia](#)

Reticências não correspondentes

Fonte e tradução não terminam com uma reticências

Verifica se as reticências ao final são replicadas entre a fonte e a tradução. Isto só verifica se há reticências reais (...) e não três pontos (. . .).

Uma reticência é normalmente mais agradável do que três pontos na impressão e soa melhor com o texto para a fala.

Veja também:

[Reticências na Wikipédia](#)

Ponto de exclamação não correspondente

Fonte e tradução não terminam com um ponto de exclamação

Verifica se as exclamações são replicadas entre a fonte e a tradução. A presença de pontos de exclamação também é verificada para vários idiomas onde eles não pertencem (chinês, japonês, coreano, armênio, limbu, mianmar ou Nko).

Veja também:

[Ponto de exclamação na Wikipédia](#)

Ponto final não correspondente

Fonte e tradução não terminam com um ponto final

Verifica se os pontos finais são replicados entre a fonte e a tradução. A presença de pontos finais é verificada para vários idiomas onde eles não pertencem (chinês, japonês, devanágari ou urdu).

Veja também:

[Ponto final na Wikipédia](#)

Ponto de interrogação não correspondente

A fonte e a tradução não terminam ambas com um ponto de interrogação

Verifica se os pontos de interrogação são replicados entre a fonte e a tradução. A presença de pontos de interrogação também é verificada para vários idiomas onde eles não pertencem (armênio, árabe, chinês, coreano, japonês, etíope, vai ou copta).

Veja também:

[Ponto de interrogação na Wikipédia](#)

Ponto e vírgula não correspondente

Fonte e tradução não terminam com ponto e vírgula

Verifica se os caracteres de ponto e vírgula no final das frases são replicados entre a fonte e a tradução. Isto pode ser útil para manter a formatação das entradas, como ficheiros de desktop.

Veja também:

[Ponto e vírgula na Wikipédia](#)

Quebras de linha não coincidentes

Quantidades de novas linhas na tradução não corresponde à fonte

Geralmente, as novas linhas são importantes para formatar a saída do programa. A verificação falha se a quantidade de literais `\n` na tradução não corresponder à fonte.

Faltam plurais

Algumas formas no plural não estão traduzidas

Verifica se todas as formas plurais de uma cadeia fonte foram traduzidas. Especificações sobre como cada forma plural é usada encontram-se na definição da cadeia.

O não preenchimento correto das formas de plural vai levar, em alguns casos, a exibir nada quando a forma do plural está em uso.

Espaços reservados

Falta a tradução de alguns espaços reservados:

Novo na versão 3.9.

Alterado na versão 4.3: Pode usar expressões regulares como espaço reservado.

A tradução não contém alguns espaços reservados. Estes são extraídos do ficheiro de tradução ou definidos manualmente a usar o sinalizador `placeholders`, outros podem ser separados com ponto e vírgula, cadeias com o espaço podem ser postos entre aspas:

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

Caso tenha alguma sintaxe para os espaços reservados, pode usar uma expressão regular:

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

Veja também:

[Personalizando comportamento a usar marcadores](#)

Espaçamento da pontuação

Espaço rígido antes do sinal de pontuação dupla

Novo na versão 3.9.

Verifica se há espaço rígido antes do sinal de pontuação dupla (ponto de exclamação, ponto de interrogação, ponto e vírgula e dois pontos). Esta regra é usada apenas em alguns idiomas selecionados, como o francês ou o bretão, onde o espaço antes do sinal de pontuação dupla é uma regra tipográfica.

Veja também:

[Espaçamento em francês e inglês na Wikipédia](#)

Expressão regular

A tradução não corresponde à expressão regular:

Novo na versão 3.9.

A tradução não corresponde à expressão regular. A expressão é extraída do ficheiro de tradução ou definida manualmente a usar o sinalizador `regex`:

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

Mesmos plurais

Algumas formas de plural estão traduzidas da mesma forma

Verifica que falha se algumas formas de plural estiverem duplicadas na tradução. Na maioria dos idiomas, elas têm que ser diferentes.

Nova linha no início

Fonte e tradução não iniciam com uma nova linha

As novas linhas costumam aparecer nas cadeias fonte por uma boa razão, de forma que omissões ou adições podem levar a problemas de formatação quando o texto traduzido é posto em uso.

Veja também:

[Nova linha no final](#)

Espaços no início

Fonte e tradução não iniciam com a mesma quantidade de espaços

Um espaço no início de uma cadeia é normalmente usado para recuo na interface, portanto, importante manter.

Nova linha no final

Fonte e tradução não terminam com uma nova linha

As novas linhas costumam aparecer nas cadeias fonte por uma boa razão, de forma que omissões ou adições podem levar a problemas de formatação quando o texto traduzido é posto em uso.

Veja também:

[Nova linha no início](#)

Espaço no final

Fonte e tradução não terminam com um espaço

Verifica se os espaços no final são replicados entre a fonte e a tradução.

O espaço ao final é normalmente utilizado para espaçar os elementos vizinhos, de forma que a sua remoção pode quebrar a disposição.

Tradução inalterada

A fonte e a tradução são idênticas

Acontece se as cadeias fonte e a tradução correspondente forem idênticos, até pelo menos uma das formas de plural. Algumas cadeias normalmente encontrados em todos os idiomas são ignorados e várias marcações são ignoradas. Isto reduz a quantidade de falsos positivos.

Esta verificação pode ajudar a encontrar cadeias erroneamente não traduzidos.

O comportamento padrão desta verificação é excluir palavras da lista negra embutida da verificação. Estas são palavras que frequentemente não estão a ser traduzidas. Isto é útil para evitar falsos positivos em cadeias curtas, que consistem apenas numa única palavra que é a mesma em vários idiomas. Esta lista negra pode ser desativada a adicionar `strict-same` à cadeia ou componente.

Veja também:

Configuração de componente, Personalizando comportamento a usar marcadores

HTML inseguro

A tradução usa uma marcação HTML insegura

Novo na versão 3.9.

A tradução utiliza uma marcação HTML insegura. Esta verificação tem que ser ativada a utilizar o sinalizador `safe-html` (veja *Personalizando comportamento a usar marcadores*). Há também o autocorretor acompanhado que pode automaticamente corrigir a marcação.

Veja também:

A verificação HTML é feita pela biblioteca [Bleach](#) desenvolvida pela Mozilla.

URL

A tradução não contém uma URL

Novo na versão 3.5.

A tradução não contém uma URL. Isso só é acionado caso a unidade esteja marcada como a conter uma URL. Nesse caso, a tradução tem de ser uma URL válido.

Markup XML

As tags na tradução não correspondem com a fonte

Isso geralmente significa que a saída resultante terá uma aparência diferente. Na maioria dos casos, este não é o resultado desejado de alterar a tradução, mas ocasionalmente é.

Verifica se as tags XML são replicadas entre a fonte e a tradução.

Sintaxe XML

A tradução não é um XML válido

Novo na versão 2.8.

A marcação XML não é válida.

Espaçamento nulo

A tradução contém um caractere extra de espaço com largura zero

Os caracteres de largura zero (<U+200B>) são usados para quebrar mensagens dentro das palavras (*word wrapping*).

Como geralmente são inseridas por engano, esta verificação é acionada uma vez que estão presentes na tradução. Alguns programas podem ter problemas quando este caractere é usado.

Veja também:

[Espaço de largura zero na Wikipédia](#)

1.6.4 Verificações de texto fonte

Estas verificações podem ajudar os programadores a melhorar a qualidade das cadeias fonte.

Reticências

A cadeia usa três pontos (...) ao invés do caractere de reticências (...)

Isto falha quando a cadeia usa três pontos (. . .) quando deveria usar um caractere de reticências (...).

O uso do caractere Unicode é, na maioria dos casos, a melhor abordagem e parece melhor renderizado e pode soar melhor com texto para fala (*text-to-speech*).

Veja também:

[Reticências na Wikipédia](#)

Não traduzido há muito tempo

Esta cadeia não foi traduzida há muito tempo

Novo na versão 4.1.

Quando a cadeia não é traduzida há muito tempo, pode indicar um problema numa cadeia fonte, a dificultar a tradução.

Várias verificações falhadas

As traduções em diversos idiomas têm verificações com falha

Várias traduções desta cadeia têm falhas nas verificações de qualidade. Isto é normalmente uma indicação de que algo poderia ser feito para melhorar a cadeia fonte.

Esta falha na verificação pode muitas vezes ser causada por um ponto final de uma frase, ou por questões menores similares que os tradutores tendem a corrigir na tradução, enquanto seria melhor corrigi-la na cadeia fonte.

Várias variáveis sem nome

Há várias variáveis sem nome na cadeia, o que impossibilita aos tradutores reordená-las

Novo na versão 4.1.

Há várias variáveis sem nome na cadeia, o que impossibilita aos tradutores reordená-las.

Considere usar variáveis nomeadas para permitir que os tradutores as reordenem.

Não pluralizado

A cadeia é usado como plural, mas não está a usar formas de plural

A cadeia é usado como um plural, mas não usa formas de plural. Caso o seu sistema de tradução tenha suporte a isso, deve usar a variante que reconheça o plural.

Por exemplo, com o Gettext em Python poderia ser:

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

1.7 Pesquisando

Novo na versão 3.9.

Consultas avançadas a usar operações booleanas, parênteses ou pesquisas específicas de campo podem ser usadas para encontrar as cadeias que deseja.

Quando nenhum campo é definido, a pesquisa acontece nos campos *Fonte*, *Tradução* e *Contexto*.

Search

All strings ▾ Sort By ▾

Advanced query builder

Source strings ▾ Search for... ☐ Exact String has suggestion ▾

String changed after ▾ mm/dd/yyyy

Query examples

Review strings changed by other users	changed:>=2021-05-16 AND NOT changed_by:testuser	<input type="button" value="Add"/>
Translated strings	state:>=translated	<input type="button" value="Add"/>
Strings with comments	has:comment	<input type="button" value="Add"/>
Strings with any failing checks	has:check	<input type="button" value="Add"/>
Strings with suggestions from others	has:suggestion AND NOT suggestion_author:testuser	<input type="button" value="Add"/>
Approved strings with suggestions	state:approved AND has:suggestion	<input type="button" value="Add"/>
All untranslated strings added the past month	added:>=2021-05-16 AND state:<=needs-editing	<input type="button" value="Add"/>
Translated strings in a certain language	is:translated AND language:cs	<input type="button" value="Add"/>

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.7.1 Pesquisa simples

Qualquer frase digitada na caixa de pesquisa é dividida em palavras. As cadeias que contêm qualquer uma delas são mostradas. Para procurar uma frase exata, ponha «a frase de pesquisa» entre aspas (ambas aspas simples (") e duplas («») funcionarão: "este é uma cadeia entre aspas" ou 'outra cadeia entre aspas'.

1.7.2 Campos

source:TEXTO Pesquisa na cadeia fonte sem diferenciar maiúscula de minúscula.

target:TEXTO Pesquisa na cadeia de destino sem diferenciar maiúscula de minúscula.

context:TEXTO Pesquisa na cadeia de contexto sem diferenciar maiúscula de minúscula.

key:TEXTO Pesquisa na cadeia chave sem diferenciar maiúscula de minúscula.

note:TEXTO Pesquisa na cadeia de comentário sem diferenciar maiúscula de minúscula.

location:TEXTO Pesquisa na cadeia de localização sem diferenciar maiúscula de minúscula.

priority:NÚMERO Prioridade da cadeia.

added:DATA E HORA Data e hora de quando a cadeia foi adicionada ao Weblate.

state:TEXTO Pesquisa de estado (approved, translated, needs-editing, empty, read-only), suporta *Operações de campo*.

pending:BOOLEANO Cadeia pendente para descarga para o VCS.

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

is:TEXTO Pesquisa por estados de cadeia (pending, translated, untranslated).

language:TEXTO Idioma de destino da cadeia.

component:TEXTO Component slug or name case insensitive search, see [URL amigável do componente](#) and [Nome do componente](#).

project:TEXTO Slug do projeto, veja [URL amigável](#).

changed_by:TEXTO A cadeia foi alterada pelo autor com o nome de utilizador fornecido.

changed:DATA E HORA O conteúdo da cadeia foi alterado na data, suporta operadores de pesquisa.

change_time:DATA E HORA A cadeia foi alterada na data, suporta [Operações de campo](#); ao contrário de `changed`, isso inclui eventos que não alteram o conteúdo e pode aplicar uma filtragem de ação personalizada a utilizar `change_action`.

change_action:TEXTO Filtros em ação de alteração, útil juntamente com `change_time`. Aceita o nome em inglês da ação de alteração, seja entre aspas e com espaços ou letras minúsculas e espaços substituídos por um hífen. Veja [Pesquisando por alterações](#) para exemplos.

check:TEXTO A cadeia tem verificação com falha.

dismissed_check:TEXTO A cadeia tem verificação dispensada.

comment:TEXTO Pesquisa nos comentários de utilizadores.

comment_author:TEXTO Filtra por autor do comentário.

suggestion:TEXTO Pesquisa nas sugestões.

suggestion_author:TEXTO Filtra por autor da sugestão.

explanation:TEXTO Pesquisa em explicações.

1.7.3 Operadores booleanos

Pode combinar pesquisas a usar AND, OR, NOT e parênteses para formar consultas complexas. Por exemplo:
`state:translated AND (source:hello OR source:bar)`

1.7.4 Operações de campo

Pode especificar operadores, intervalos ou pesquisas parciais para pesquisas de data ou numéricas:

state:>=translated Estado é `translated` ou melhor (`approved`).

changed:2019 Alterado no ano de 2019.

changed:[2019-03-01 to 2019-04-01] Alterado entre as duas datas dadas.

1.7.5 Operadores exatos

Pode fazer uma consulta de correspondência exata em diferentes campos de texto a usar o operador `=`. Por exemplo, para procurar por todas as cadeias fonte que correspondem exatamente a `hello world`, use: `source:="hello world"`. Para pesquisar por expressões de uma única palavra, não precisa de aspas. Por exemplo, para procurar por todas as cadeias fonte que correspondam a `hello`, pode utilizar: `source:=hello`.

1.7.6 Pesquisando por alterações

Novo na versão 4.4.

A pesquisa por eventos do histórico pode ser feita a utilizar os operadores `change_action` e `change_time`.

Por exemplo, uma pesquisa por cadeias marcadas para edição em 2018 pode ser inserida como `change_time:2018 AND change_action:marked-for-edit` ou `change_time:2018 AND change_action:"Marked for edit"`.

1.7.7 Expressões regulares

Em qualquer lugar que o texto seja aceito, também pode especificar uma expressão regular como `r "regexp"`.

Por exemplo, para pesquisar por todas as cadeias que contenham qualquer dígito entre 2 e 5, utilize `source:r"[2-5]"`.

1.7.8 Consultas predefinidas

Pode seleccionar entre as consultas predefinidas na página de pesquisa, o que lhe permite acessar rapidamente as pesquisas mais frequentes:

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

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

translated 96%

1/1

<

>

Custom search

Zen

Translation

English

Singular
 %(count)s word

Plural
 %(count)s words

Czech, One

%(count)s slovo

Czech, Few

%(count)s slova

Czech, Other

%(count)s slov

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

☐ Needs editing

Save

Suggest

Skip

Not translated strings • state:empty

Strings needing action • 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

Strings needing action 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

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.h tml:149

String age

7 seconds ago

Source string age

7 seconds ago

Translation file

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

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Powered by Weblate 4.7
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

1.7. Pesquisando

47

1.7.9 Ordenando os resultados

Há muitas opções para ordenar as cadeias conforme as suas necessidades:

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area displays a translation for the string 'The string uses three dots (...) instead of an ellipsis character (...)'. Below this, there are buttons for 'Save', 'Suggest', and 'Skip'. A dropdown menu is open, showing various sorting options: 'Position and priority', 'Position', 'Priority', 'Labels', 'Source string', 'Translated string', 'Age of string', 'Number of words', 'Number of comments', 'Number of failing checks', and 'Key'. The 'Position and priority' option is selected. On the right side, there is a sidebar with information about the string, including 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

1.8 Fluxos de trabalho de tradução

Usar o Weblate é um processo que lhe aproxima os seus utilizadores, a aproximar-o dos seus tradutores. Depende de decidir quantos recursos deseja usar.

A lista a seguir não é uma lista completa de maneiras de configurar o Weblate. Pode basear outros fluxos de trabalho nos exemplos mais usuais listados aqui.

1.8.1 Acesso à tradução

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.

Nos capítulos a seguir, *qualquer utilizador* significa um utilizador que tenha acesso à tradução. Pode ser qualquer utilizador autenticado se o projeto for público ou um utilizador que tenha uma permissão *Traduzir* para o projeto.

1.8.2 Estados de tradução

Cada cadeia traduzida pode estar num dos seguintes estados:

Não traduzido A tradução está vazia, pode ou não estar armazenada no ficheiro, dependendo do formato do ficheiro.

Precisa de edição A tradução precisa ser editada, isso geralmente é o resultado de uma alteração de cadeias fonte, correspondência aproximada ou ação de um tradutor. A tradução está armazenada no ficheiro, a depender do formato do ficheiro que pode ser marcado como necessidade de edição (por exemplo, da mesma forma que receberia uma sinalização `fuzzy` no ficheiro de Gettext).

A aguardar por revisão A tradução está feita, mas não revisada. É armazenada no ficheiro como uma tradução válida.

Aprovadas A tradução foi aprovada na revisão. Já não pode ser alterada por tradutores, mas apenas por revisores. Tradutores só podem adicioná-las sugestões.

Sugestões As sugestões estão armazenadas apenas no Weblate e não no ficheiro de tradução.

Os estados são representados nos ficheiros de tradução quando possível.

Dica: In case file format you use does not support storing states, you might want to use *Marcar as traduções inalteradas como «Precisa de edição»* addon to flag unchanged strings as needing editing.

Veja também:

Capacidades dos tipos de tradução, Fluxos de trabalho de tradução

1.8.3 Tradução direta

Esta é a configuração mais usual para equipas menores, qualquer um pode traduzir diretamente. Esta também é a configuração predefinida no Weblate.

- *Qualquer utilizador* pode editar traduções.
- Sugestões são formas opcionais de sugerir alterações, quando os tradutores não têm certeza sobre a alteração.

Configuração	Valor	Nota
Activar revisões	inativo	Configurado a nível de projeto.
Ativar sugestões	ativo	É útil para os utilizadores serem capazes de sugerir quando não têm certeza.
Votação de sugestão	inativo	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	<i>Utilizadores</i>	Or <i>Translate with per-project access control.</i>
Grupo de revisores	N/D	Não usado.

1.8.4 Revisão por pares

Com este fluxo de trabalho, qualquer pessoa pode adicionar sugestões e precisa da aprovação de um ou mais membros adicionais antes de ser aceite como tradução.

- *Qualquer utilizador* pode adicionar sugestões.
- *Qualquer utilizador* pode votar em sugestões.
- Sugestões tornam-se traduções quando dado uma quantidade predeterminada de votos.

Configuração	Valor	Nota
Activar revisões	inativo	Configurado a nível de projeto.
Ativar sugestões	ativo	
Votação de sugestão	inativo	
Aceitar sugestões automaticamente	1	Pode definir um valor mais alto para exibir mais revisões por pares.
Grupo de tradutores	<i>Utilizadores</i>	Or <i>Translate with per-project access control.</i>
Grupo de revisores	N/D	Não usado, todos os tradutores revisam.

1.8.5 Revisores dedicados

Novo na versão 2.18: O fluxo de trabalho adequado de revisão é suportado desde o Weblate 2.18.

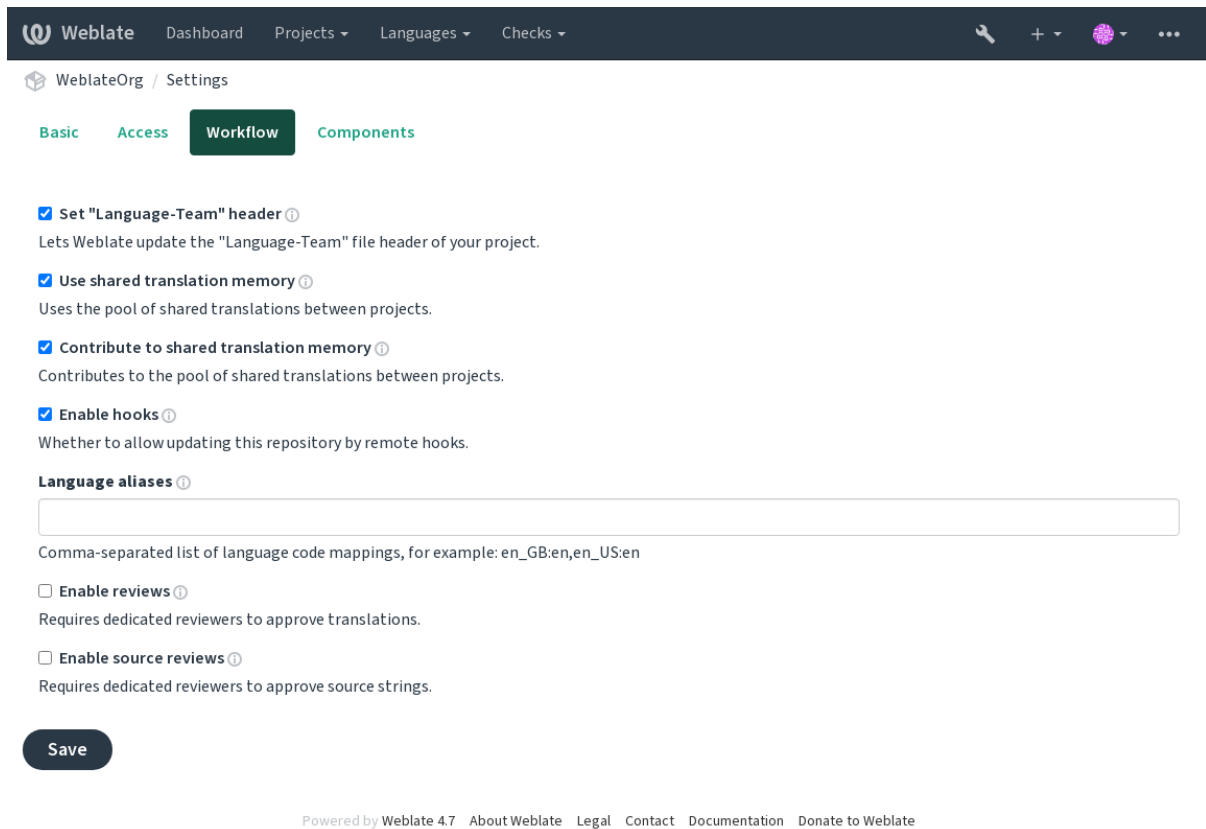
Com revisores dedicados tem dois grupos de utilizadores, um capaz de enviar traduções e outro capaz de revisá-los para garantir que as traduções sejam consistentes e que a qualidade seja boa.

- *Qualquer utilizador* pode editar traduções não aprovadas.
- *Revisor* pode aprovar / retirar a aprovação de cadeias.
- *Revisor* pode editar todas as traduções (incluindo as aprovadas).
- Sugestões também podem ser usadas para sugerir alterações para cadeias aprovadas.

Configuração	Valor	Nota
Activar revisões	ativo	Configurado a nível de projeto.
Ativar sugestões	inativo	É útil para os utilizadores serem capazes de sugerir quando não têm certeza.
Votação de sugestão	inativo	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	<i>Utilizadores</i>	Or <i>Translate with per-project access control.</i>
Grupo de revisores	<i>Revisores</i>	Or <i>Review with per-project access control.</i>

1.8.6 Ativar revisões

As revisões podem ser ativadas na configuração do projeto, a partir da subpágina *Fluxo de trabalho* das configurações do projeto (pode ser encontrada no menu *Gerir* → *Configurações*):

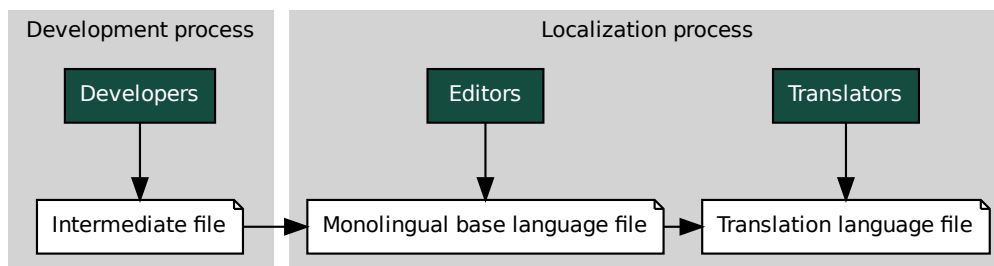


Nota: Dependendo da configuração do Weblate, a configuração pode não estar-lhe disponível. Por exemplo, no Hosted Weblate, isso não está disponível para projetos hospedados gratuitamente.

1.8.7 Portal de qualidade para cadeias fonte

Em muitos casos, as cadeias fonte do idioma de origem vêm de programadores, porque eles escrevem o código e fornecem cadeias iniciais. No entanto, os programadores muitas vezes não são falantes nativos do idioma de origem e não fornecem a qualidade desejada das cadeias fonte. A tradução intermediária pode ajudá-los a lidar com isso - há uma rota de qualidade adicional para as cadeias entre programadores e tradutores e utilizadores.

Ao definir um *Ficheiro de idioma intermédio*, este ficheiro será usado como fonte para as cadeias, mas será editado para o idioma de origem para poli-lo. Uma vez que o texto esteja pronto no idioma de origem, também estará disponível para os tradutores traduzirem em idiomas adicionais.

**Veja também:**

Ficheiro de idioma intermédio, Ficheiro de idioma base monolíngue, Formatos bilíngues e monolíngues

1.8.8 Revisões de cadeias fonte

Com o *Ativar revisões de fontes* ativado, o processo de revisão pode ser aplicado em cadeias fonte. Uma vez ativado, os utilizadores podem relatar problemas nas cadeias fonte. O processo real depende se usa formatos bilíngues ou monolíngues.

Para formatos monolíngues, a revisão de cadeias fonte se comporta da mesma forma que com *Revisores dedicados* - uma vez que o problema é relatado na cadeia fonte, é marcado como *Necessita edição*.

Os formatos bilíngues não permitem a edição direta de cadeias fonte (estes são normalmente extraídos diretamente do código-fonte). Neste caso, o rótulo *Fonte precisa de revisão* é anexado às cadeias relatadas por tradutores. Deve revisar esses textos e editá-los na fonte ou remover o rótulo.

Veja também:

Formatos bilíngues e monolíngues, Revisores dedicados, labels, Comentários

1.9 Perguntas frequentes

1.9.1 Configuração

Como criar um fluxo de trabalho automatizado?

Weblate pode tratar de todas as coisas de tradução semiautomaticamente para si. Se der acesso de push ao seu repositório, as traduções podem acontecer sem interação, a menos que algum conflito de mesclagem ocorra.

1. Configure o seu repositório Git para informar Weblate quando houver alguma alteração, veja *Hooks de notificação* para informações sobre como fazer isso.
2. Defina uma URL de push na sua *Configuração de componente* no Weblate, isto permite que o Weblate faça o push das alterações para o seu repositório.
3. Turn on *Enviar ao submeter* on your *Configuração de componente* in Weblate, this will make Weblate push changes to your repository whenever they happen at Weblate.

Veja também:

Tradução contínua, Evitar conflitos de mesclagem

Como acessar os repositórios por SSH?

Por favor, veja *Acessando repositórios* para informações sobre a configuração das chaves SSH.

Como resolver conflitos de mesclagem nas traduções?

Os conflitos de mesclagem acontecem de tempos em tempos quando o ficheiro de tradução é alterado tanto na Weblate como no repositório upstream ao mesmo tempo. Geralmente pode evitar-lo a mesclar traduções Weblate antes de fazer alterações nos ficheiros de tradução (por exemplo, antes de executar o msgmerge). Apenas diga ao Weblate para fazer o commit de todas as traduções pendentes (pode fazê-lo em *Manutenção do repositório* no menu *Gerir*) e mesclar o repositório (se o push automático não estiver ligado).

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 *Exportador git* 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
```

Se estiver a usar vários ramos na Weblate, pode fazer o mesmo com todos eles:

```
# 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
```

(continues on next page)

(continuação da página anterior)

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

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

No caso de ficheiros PO gettext, há uma forma de mesclar conflitos de uma forma semiautomática:

Obtenha e mantenha um clone local do repositório Weblate Git. Obtenha também um segundo clone local novo do repositório upstream Git (ou seja, precisa de duas cópias do repositório upstream Git: Uma cópia intacta e uma cópia de trabalho):

```
# 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
```

Veja também:

Como exportar o repositório Git que a Weblate usa?, Tradução contínua, Evitar conflitos de mesclagem, Cliente Weblate

Como faço para traduzir vários ramos ao mesmo tempo?

Weblate tem suporte a fazer push de alterações de tradução dentro de uma *Configuração de projeto*. Para cada *Configuração de componente* que o tem ligado (o comportamento padrão), a alteração feita é automaticamente propagada para outros. Desta forma as traduções são mantidas sincronizadas mesmo que os próprios ramos já tenham divergido bastante e não é possível simplesmente fundir as mudanças de tradução entre eles.

Uma vez mescladas as mudanças do Weblate, pode ter que mesclar esses ramos (a depender do seu fluxo de trabalho de desenvolvimento) a descartar as diferenças:

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

Veja também:

Manter traduções iguais entre componentes

Como traduzir projetos multiplataforma?

Weblate tem suporte a uma grande variedade de formatos de ficheiro (veja [Formatos de ficheiros suportados](#)) e a abordagem mais fácil é utilizar o formato nativo para cada plataforma.

Dado que tenha adicionado todos os ficheiros de tradução da plataforma como componentes num projeto (veja [Adicionando projetos e componentes de tradução](#)), pode utilizar o recurso de propagação de tradução (ligado por padrão, e pode ser desligado na [Configuração de componente](#)) para traduzir cadeias para todas as plataformas ao mesmo tempo.

Veja também:

Manter traduções iguais entre componentes

Como exportar o repositório Git que a Weblate usa?

Não há nada de especial no repositório, ele vive sob o diretório `DATA_DIR` e é chamado `vcs/<projeto>/<componente>/`. Se tem acesso SSH a esta máquina, pode usar o repositório diretamente.

Para acesso anônimo, querará executar um servidor Git e deixá-lo servir o repositório para o mundo exterior.

Alternativamente, pode usar [Exportador git](#) dentro do Weblate para automatizar isto.

Quais são as opções para fazer push das alterações de volta para o upstream?

Isto depende muito da sua configuração, Weblate é bastante flexível nesta área. Aqui estão exemplos de alguns fluxos de trabalho usados com o Weblate:

- O Weblate faz push e mescla automaticamente alterações (veja [Como criar um fluxo de trabalho automatizado?](#)).
- Diz manualmente ao Weblate para fazer push (ele precisa de acesso push ao repositório upstream).
- Alguém mescla manualmente as alterações do repositório Weblate git para o repositório upstream.
- Alguém reescreve o histórico produzido pelo Weblate (por exemplo, a eliminar compromissos de mesclagem), mescla alterações e diz ao Weblate para redefinir o conteúdo no repositório upstream.

Claro que é livre para misturar tudo isso como quiser.

Como posso limitar o acesso ao Weblate apenas às traduções, sem lhe expor o código-fonte?

Pode utilizar o comando `git submodule` para separar as traduções do código-fonte enquanto ainda as tem sob controle de versão.

1. Crie um repositório com os seus ficheiros de tradução.
2. Adicione-o como um submódulo ao seu código:

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

3. Vincule o Weblate a este repositório, ele não precisa mais ter acesso ao repositório que contém o seu código-fonte.
4. Pode atualizar o repositório principal com traduções do Weblate a executar:

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

Por favor, consulte a documentação do submódulo `git` para obter mais detalhes.

Como posso verificar se a minha Weblate está configurada corretamente?

Weblate inclui um conjunto de verificações de configuração que pode ver na interface administrativa, basta seguir a ligação *Relatório de desempenho* na interface administrativa, ou abrir a URL `/manage/performance/` diretamente.

Por que é que todos os commits são feitos por Weblate <noreply@weblate.org>?

Este é o nome padrão do «commiter», configurado quando cria um componente de tradução. Pode alterá-lo na administração a qualquer momento.

O autor de cada commit (se o VCS subjacente o suportar) ainda é registado corretamente como o utilizador que fez a tradução.

Veja também:

Configuração de componente

1.9.2 Uso

Como faço para revisar as traduções dos outros?

- Há vários fluxos de trabalho baseados em revisões disponíveis em Weblate, veja *Fluxos de trabalho de tradução*.
- Pode se inscrever para quaisquer alterações feitas em *Notificações* e depois verificar outras contribuições à medida que elas chegam por e-mail.
- Há uma ferramenta de revisão disponível na parte inferior da vista de tradução, onde pode optar por navegar pelas traduções feitas por outros desde uma data determinada.

Veja também:

Fluxos de trabalho de tradução

Como posso fornecer feedback sobre uma cadeia fonte?

Nas abas de contexto abaixo da tradução, pode utilizar a guia *Comentários* para fornecer feedback sobre uma cadeia fonte, ou discuti-lo com outros tradutores.

Veja também:

report-source, *Comentários*

Como posso usar as traduções existentes enquanto traduzo?

- Todas as traduções no Weblate podem ser usadas graças à memória de tradução compartilhada.
- Pode importar ficheiros de memória de tradução existentes para o Weblate.
- Use a funcionalidade de importação para carregar compêndios como traduções, sugestões ou traduções que necessitem de revisão. Esta é a melhor abordagem para uma tradução única a usar um compêndio ou uma base de dados de tradução similar.
- Pode configurar *tmserver* com todos os bancos de dados que tem e deixar o Weblate utilizá-lo. Isto é bom quando quer utilizá-lo várias vezes durante a tradução.
- Outra opção é traduzir todos os projetos relacionados numa única instância Weblate, o que o fará pegar automaticamente as traduções de outros projetos também.

Veja também:

Tradução automática, Sugestões automáticas, Memória de Tradução

O Weblate atualiza ficheiros de tradução além de traduções?

Weblate tenta limitar ao mínimo as alterações nos ficheiros de tradução. Para alguns formatos de ficheiro, infelizmente, isso pode levar a reformatar o ficheiro. Se quiser manter o ficheiro formatado do seu jeito, use um gancho (hook) de pré-commit para isso.

Veja também:

updating-target-files

De onde vêm as definições linguísticas e como posso adicionar as minhas?

O conjunto básico de definições de idiomas está incluído no Weblate e no Translate-toolkit. Isso abrange mais de 150 idiomas e inclui informações sobre formas plurais ou direção do texto.

É livre para definir os seus próprios idiomas na interface administrativa, só precisa fornecer informações sobre isso.

Veja também:

Definições de idioma

Weblate pode destacar alterações numa cadeia aproximada?

Weblate tem suporte a isso, porém precisa dos dados para mostrar a diferença.

Para ficheiros de PO Gettext, tem que passar o parâmetro `--previous` para **msgmerge** ao atualizar ficheiros de PO, por exemplo:

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

Para traduções monolíngues, a Weblate pode encontrar a cadeia anterior por ID, por isso mostra as diferenças automaticamente.

Por que é que o Weblate ainda mostra as cadeias de tradução antigas quando atualizei o modelo?

Weblate não tenta manipular os ficheiros de tradução de nenhuma outra forma que não seja permitir que os tradutores traduzam. Por isso, também não atualiza os ficheiros traduzíveis quando o modelo ou código-fonte tiverem sido alterados. Simplesmente tem que fazer isso manualmente e fazer push das alterações para o repositório, o Weblate irá então pegar as alterações automaticamente.

Nota: Normalmente é uma boa ideia mesclar as alterações feitas na Weblate antes de atualizar os ficheiros de tradução, pois de outra forma geralmente acabará com alguns conflitos para mesclar.

Por exemplo, com os ficheiros PO gettext, pode atualizar os ficheiros de tradução a utilizar a ferramenta **msgmerge**:

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

Caso queira fazer a atualização automaticamente, pode instalar a extensão *Atualizar ficheiros PO para coincidir com POT (msgmerge)*.

Veja também:

updating-target-files

1.9.3 Soluções de problemas

As solicitações às vezes falham com o erro «demais ficheiros abertos»

Isto acontece às vezes quando o seu repositório Git cresce demais e tem muitos deles. Comprimir os repositórios Git irá melhorar esta situação.

A maneira mais fácil de fazer isto é executar:

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

Veja também:

DATA_DIR

Ao acessar ao site recebo um erro de «Bad Request (400)»

Isto é muito provavelmente causado por uma *ALLOWED_HOSTS* mal configurada. Ele precisa conter todos os nomes de host que quer acessar no seu Weblate. Por exemplo:

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

Veja também:

Configuração de hosts permitidos

O que significa «Existem mais ficheiros para o idioma único (en)»?

Isso normalmente acontece quando tem um ficheiro de tradução para o idioma fonte. Weblate mantém o controlo de cadeias fonte e reserva o idioma fonte para isso. O ficheiro adicional para o mesmo idioma não é processado.

- Se a tradução para o idioma de origem for desejada, por favor altere o *Idioma fonte* nas configurações dos componentes.
- Caso o ficheiro de tradução para o idioma de origem não seja necessário, por favor, remova-o do repositório.
- Caso o ficheiro de tradução para o idioma de origem seja necessário, mas deveria ser ignorado pelo Weblate, por favor, ajuste o filtro do idioma para excluí-lo.

Dica: Também pode receber mensagens de erro semelhantes para outros idiomas. Nesse caso, o motivo mais provável é que vários ficheiros mapeiam para um único idioma no Weblate.

Isto pode ser causado pelo uso de códigos de idioma obsoletos juntamente com um novo (j_a e j_p para japonês) ou incluindo tanto códigos específicos do país como genéricos (fr e fr_FR). Veja language-parsing-codes para mais detalhes.

1.9.4 Funcionalidades

O Weblate tem suporte a outros VCSs que não o Git e o Mercurial?

Weblate atualmente não tem suporte nativo para nada além de *Git* (com suporte estendido para *GitHub*, vcs-gerrit” e :ref:`vcs-git-svn`) e *Mercurial*, mas é possível escrever backends para outros VCSs.

Também pode usar *Auxiliares de remotos do Git* no Git para acessar outros VCSs.

Weblate also supports VCS-less operation, see *Ficheiros locais*.

Nota: Para suporte nativo de outros VCSs, Weblate requer o uso de VCS distribuídos e provavelmente poderia ser ajustado para trabalhar com qualquer outra coisa além de Git e Mercurial, mas alguém tem que implementar esse suporte.

Veja também:

Integração de controlo de versões

Como o Weblate credita os tradutores?

Cada alteração feita no Weblate é posta em commit no VCS sob o nome dos tradutores. Desta forma, cada alteração tem a devida autoria e pode rastreá-la a usar as ferramentas VCS padrão que usa para código.

Além disso, quando o formato do ficheiro de tradução o suporta, os cabeçalhos do ficheiro são atualizados para incluir o nome do tradutor.

Veja também:

list_translators, *../devel/reporting*

Por que o Weblate força mostrar todos os ficheiros PO numa única árvore?

O Weblate foi projetado de uma forma que cada ficheiros PO é representado como um único componente. Isso é benéfico para os tradutores, então eles sabem o que estão realmente a traduzir.

Alterado na versão 4.2: Tradutores podem traduzir todos os componentes de um projeto num idioma específico como um todo.

Por que o Weblate usa códigos de idioma como sr_Latn ou 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).

O Weblate ainda entende códigos de linguagem legados e irá mapeá-los para o atual - por exemplo sr@latin será tratado como sr_Latn ou zh@CN como zh_Hans.

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

Veja também:

Definições de idioma, *Estilo de código de idioma*, *Adicionando novas traduções*

1.10 Formatos de ficheiros suportados

O Weblate tem suporte à maioria dos formatos de tradução compreendidos pelo [translate-toolkit](#), porém cada formato a ser ligeiramente diferente, alguns problemas com formatos que não são bem testados podem surgir.

Veja também:

[Translation Related File Formats](#)

Nota: Ao escolher um formato de ficheiro para a sua aplicação, é melhor pôr algum formato bem estabelecido no kit de ferramentas/plataforma que usa. Dessa forma, os seus tradutores podem usar adicionalmente todas as ferramentas que estão acostumados e provavelmente contribuirão para o seu projeto.

1.10.1 Formatos bilíngues e monolíngues

Ambos os formatos monolíngue e bilingue são suportados. Formatos bilíngues armazenam dois idiomas num único ficheiro e tradução (exemplos típicos são *GNU gettext*, *XLIFF* ou *Cadeias de Apple iOS*). Por outro lado, formatos monolíngues identificam a cadeia por ID e cada ficheiro de idioma contém apenas o mapeamento desses para qualquer idioma (tipicamente *Recurso de cadeias de Android*). Alguns formatos de ficheiro são utilizados em ambas as variantes, veja a descrição detalhada abaixo.

Para o uso correto de ficheiros monolíngues, Weblate requer acesso a um ficheiro a conter uma lista completa de cadeias para traduzir com a fonte dele – este ficheiro é chamado de *Ficheiro de idioma base monolíngue* dentro do Weblate, embora a nomenclatura possa variar no seu paradigma.

Além disso, este fluxo de trabalho pode ser estendido a usar *Ficheiro de idioma intermédio* para incluir cadeias fornecidas pelos programadores, mas não para ser usado como está nos textos finais.

1.10.2 Detecção automática

O Weblate pode detetar automaticamente vários formatos de ficheiros generalizados, mas essa detecção pode prejudicar o seu desempenho e limitará recursos específicos para o formato de ficheiro dado (por exemplo, a adição automática de novas traduções).

1.10.3 Capacidades dos tipos de tradução

Capacidades de todos os formatos suportados:

Formato	Linguali- dade <small>Página 62, 1</small>	Plu- rais <small>Página 62, 2</small>	Comentá- rios <small>Página 62, 3</small>	Con- texto <small>Página 62, 4</small>	Localiza- ção <small>Página 62, 5</small>	Marcado- res <small>Página 62, 8</small>	Estados adicio- nais <small>Página 62, 6</small>
<i>GNU gettext</i>	bilíngue	sim	sim	sim	sim	sim ⁹	necessita edição
<i>Gettext mo- nolíngue</i>	mono	sim	sim	sim	sim	sim ⁷	necessita edição
<i>XLIFF</i>	ambos	sim	sim	sim	sim	sim ¹⁰	necessita edição, aprovado
<i>Propri- edades Java</i>	ambos	não	sim	não	não	não	
<i>mi18n lang files</i>	mono	não	sim	não	não	não	

continues on next page

Table 1 – continuação da página anterior

Formato	Linguali- dade ^{Página 62, 1}	Plu- rais ^{Página 62, 2}	Comentá- rios ^{Página 62, 3}	Con- texto ^{Página 62, 4}	Localiza- ção ^{Página 62, 5}	Marcado- res ^{Página 62, 8}	Estados adicio- nais ^{Página 62, 6}
<i>Propriedades GWT</i>	mono	sim	sim	não	não	não	
<i>Traduções para Joomla</i>	mono	não	sim	não	sim	não	
<i>.ts de Qt Linguist</i>	ambos	sim	sim	não	sim	sim ⁷	necessita edição
<i>Recurso de cadeias de Android</i>	mono	sim	sim ⁷	não	não	sim ⁷	
<i>Cadeias de Apple iOS</i>	bilíngue	não	sim	não	não	não	
<i>Cadeias de PHP</i>	mono	não ¹¹	sim	não	não	não	
<i>Ficheiro JSON</i>	mono	não	não	não	não	não	
<i>Ficheiros i18next do JSON</i>	mono	sim	não	não	não	não	
<i>Ficheiros JSON go-i18n</i>	mono	sim	não	não	não	não	
<i>Ficheiro ARB</i>	mono	sim	sim	não	não	não	
<i>JSON We- bExtension</i>	mono	sim	sim	não	não	não	
<i>Ficheiros de recursos .XML</i>	mono	não	sim	não	não	sim ⁷	
<i>Ficheiros CSV</i>	ambos	não	sim	sim	sim	não	necessita edição
<i>Ficheiros YAML</i>	mono	não	sim	não	não	não	
<i>Ficheiros Ruby YAML</i>	mono	sim	sim	não	não	não	
<i>Ficheiros DTD</i>	mono	não	não	não	não	não	
<i>Ficheiro XML simples</i>	mono	não	não	não	não	sim ⁷	
<i>Ficheiros RC do Windows</i>	mono	não	sim	não	não	não	
<i>Excel Open XML</i>	mono	não	sim	sim	sim	não	necessita edição
<i>Ficheiros de metada- dos da App Store</i>	mono	não	não	não	não	não	

continues on next page

Table 1 – continuação da página anterior

Formato	Linguali- dade ^{Página 62, 1}	Plu- rais ^{Página 62, 2}	Comentá- rios ^{Página 62, 3}	Con- texto ^{Página 62, 4}	Localiza- ção ^{Página 62, 5}	Marcado- res ^{Página 62, 8}	Estados adicio- nais ^{Página 62, 6}
<i>Ficheiros de legenda</i>	mono	não	não	não	sim	não	
<i>Ficheiro HTML</i>	mono	não	não	não	não	não	
<i>Formato OpenDocument</i>	mono	não	não	não	não	não	
<i>Formato IDML</i>	mono	não	não	não	não	não	
<i>Traduções de INI</i>	mono	não	não	não	não	não	
<i>Traduções Inno Setup INI</i>	mono	não	não	não	não	não	
<i>Formato TermBase eXchange</i>	bilíngue	não	sim	não	não	sim ⁷	
<i>Ficheiros de texto</i>	mono	não	não	não	não	não	

Cadeias somente leitura

Novo na versão 3.10.

Cadeias de somente leitura dos ficheiros de tradução serão incluídos, mas não podem ser editadas no Weblate. Esse recurso é suportado nativamente por poucos formatos (*XLIFF* e *Recurso de cadeias de Android*), mas pode ser emulado em outros a adicionar um marcador `read-only`, consulte *Personalizando comportamento a usar marcadores*.

1.10.4 GNU gettext

O formato mais usado para traduzir software livre.

As informações contextuais armazenadas no ficheiro são suportadas a ajustar os cabeçalhos dele ou a vincular-se a ficheiros fonte correspondentes.

O ficheiro de PO de texto bilíngue normalmente parece-se com este:

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

(continues on next page)

¹ Veja *Formatos bilíngues e monolíngues*

² Os plurais são necessários para localizar adequadamente as cadeias com contagem variável.

³ Os comentários podem ser usados para passar informações adicionais sobre a cadeia a traduzir.

⁴ O contexto é usado para diferenciar cadeias idênticas usadas em diferentes escopos (por exemplo, *Sun* pode ser usado como um nome abreviado do dia «Sunday» ou como o nome da nossa estrela mais próxima).

⁵ A localização de uma cadeia no código-fonte pode ajudar tradutores proficientes a descobrir como a cadeia é usada.

⁶ Veja *Personalizando comportamento a usar marcadores*

⁷ Estados adicionais suportados pelo formato do ficheiro, além de «Não traduzido» e «Traduzido».

⁸ Os comentários do tipo gettext são usados como marcadores.

⁹ Os marcadores são extraídos do atributo não padrão `weblate-flags` para todos os formatos baseados em XML. Adicionalmente, o `max-length:N` é suportado através do atributo `maxwidth` como definido no padrão XLIFF, veja *Especificação de bandeiras de tradução*.

¹⁰ Comentário XML posto antes do elemento `<string>`, analisado como um comentário do programador.

¹¹ Os plurais são suportados apenas para Laravel, que usa em sintaxe de cadeias para defini-las, veja *Localização em Laravel*.

(continuação da página anterior)

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

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

Configuração de componente típica do Weblate

Máscara de ficheiro	po/* .po
Ficheiro de idioma base monolíngue	Vazio
Modelo para novas traduções	po/messages.pot
Formato de ficheiro	Ficheiro Gettext PO

Veja também:

devel/gettext, devel/sphinx, Gettext no Wikipédia, PO Files, *Atualizar a variável ALL_LINGUAS no ficheiro «configure»*, *Personalizar a saída gettext*, *Atualizar ficheiro LINGUAS*, *Gerar ficheiros MO*, *Atualizar ficheiros PO para coincidir com POT (msgmerge)*

Gettext monolíngue

Alguns projetos decidem usar o gettext como formatos monolíngues – eles codificam apenas os IDs no código-fonte deles e a cadeia então precisa ser traduzido para todos os idiomas, que inclui o inglês. É suportado, embora tenha que escolher este formato de ficheiro explicitamente ao importar componentes para o Weblate.

O ficheiro gettext PO monolíngue normalmente se parece com este:

```
#: 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"
```

Enquanto o ficheiro de idioma base será:

```
#: 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"
```

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	po/* .po
Ficheiro de idioma base monolíngue	po/en .po
Modelo para novas traduções	po/messages .pot
Formato de ficheiro	<i>Ficheiro Gettext PO (monolíngue)</i>

1.10.5 XLIFF

Formato baseado em XML criado para padronizar ficheiros de tradução, mas no final é um dos [muitos padrões](#), nesta área.

XML Localization Interchange File Format (XLIFF) é normalmente utilizado como bilíngue, mas Weblate também o suporta como monolíngue.

Veja também:

Especificação do `XML Localization Interchange File Format (XLIFF)

Estados de tradução

Alterado na versão 3.3: Weblate ignored the `state` attribute prior to the 3.3 release.

O atributo `state` no ficheiro é parcialmente processado e mapeado para o estado «Necessita edição» no Weblate (os seguintes estados são utilizados para marcar a cadeia como a necessitar editar se houver um alvo presente: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). Caso o atributo `state` esteja a faltar, uma cadeia é considerada traduzida assim que um elemento `<target>` existir.

Se a cadeia de tradução tiver `approved="yes"`, ela também será importada para a Weblate como «Aprovado», qualquer outra coisa será importada como «A aguardar revisão» (que corresponde à especificação XLIFF).

Ao gravar, o Weblate não adiciona esses atributos a menos que seja necessário:

- O atributo `state` só é adicionado no caso da cadeia ser marcada como a necessitar de edição.
- O atributo `approved` só é adicionado no caso da cadeia ter sido revista.
- Em outros casos os atributos não são adicionados, mas são atualizados no caso de estarem presentes.

Isso significa que, ao utilizar o formato XLIFF, é fortemente recomendado ativar o processo de revisão do Weblate, de modo a ver e alterar o estado aprovado de cadeias.

Da mesma forma, ao importar tais ficheiros (no formulário de envio), deve escolher *Importador como traduzido* em *A processar as cadeias que necessitam edição*.

Veja também:

Revisores dedicados

Espaços em branco e novas linhas em XLIFF

Geralmente os tipos ou quantidades de espaços em branco não são diferenciados entre si nos formatos XML. Se quiser mantê-los, tem que adicionar o marcador `xml:space="preserve"` à cadeia.

Por exemplo:

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

Especificação de bandeiras de tradução

Pode especificar marcadores de tradução adicionais (veja *Personalizando comportamento a usar marcadores*) a utilizar o atributo `weblate-flags`. Weblate também entende os atributos `maxwidth` e `font` da especificação XLIFF:

```
<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>
```

O atributo `font` é analisado para a família de fontes, tamanho e peso, o exemplo acima mostra tudo isso, embora apenas a família de fontes seja necessária. Qualquer espaço em branco na família de fontes é convertido para sublinhado, então `Source Sans Pro` torna-se `Source_Sans_Pro`, por favor tenha isso em mente ao nomear o grupo de fontes (veja *Gerir letras*).

Chaves de cadeias

Weblate identifica as unidades no ficheiro XLIFF pelo atributo `resname` no caso de estar presente e, do contrário, recorre a `id` (junto com a tag `file` se presente).

O atributo `resname` deveria ser um identificador humanamente amigável da unidade, a tornar-a mais adequada para a exibição de Weblate em vez de `id`. O `resname` tem de ser único em todo o ficheiro XLIFF. Isto é exigido pelo Weblate e não é coberto pelo padrão XLIFF – não coloca nenhuma restrição de exclusividade neste atributo.

<i>Configuração de componente típica do Weblate para XLIFF bilíngue</i>	
Máscara de ficheiro	<code>localizations/*.xliff</code>
Ficheiro de idioma base monolíngue	<i>Vazio</i>
Modelo para novas traduções	<code>localizations/en-US.xliff</code>
Formato de ficheiro	<i>Ficheiro de tradução XLIFF</i>

<i>Configuração de componente típica do Weblate para XLIFF monolíngue</i>	
Máscara de ficheiros	<code>localizations/*.xliff</code>
Ficheiro de idioma base monolíngue	<code>localizations/en-US.xliff</code>
Modelo para novas traduções	<code>localizations/en-US.xliff</code>
Formato de ficheiro	<i>Ficheiro de tradução XLIFF</i>

Vea também:

[XLIFF na Wikipédia](#), [XLIFF](#), [Atributo font no XLIFF 1.2](#), [Atributo maxwidth no XLIFF 1.2](#)

1.10.6 Propriedades Java

Formato nativo do Java para traduções.

As propriedades Java são geralmente usadas como traduções monolíngues.

Weblate é compatível com as variantes ISO-8859-1, UTF-8 e UTF-16 deste formato. Todos eles suportam o armazenamento de todos os caracteres Unicode, mas são codificados de forma diferente. No ISO-8859-1, as sequências de escape Unicode são usadas (por exemplo, `zkou\u0161ka`), todas as outras codificam caracteres diretamente em UTF-8 ou UTF-16.

Nota: Carregar sequências de escape também funciona no modo UTF-8, por isso, tenha cuidado ao escolher o conjunto de codificação correto para corresponder às necessidades da aplicação.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	src/app/Bundle_*.properties
Ficheiro de idioma base monolíngue	src/app/Bundle.properties
Modelo para novas traduções	Vazio
Formato de ficheiro	Propriedades Java (ISO-8859-1)

Veja também:

Java properties on Wikipedia, Mozilla and Java properties files, *mi18n lang files*, *Propriedades GWT*, updating-target-files, *Formata as propriedades do ficheiro Java*, *Limpeza de ficheiros de tradução*

1.10.7 mi18n lang files

Novo na versão 4.7.

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	*.lang
Ficheiro de idioma base monolíngue	en-US.lang
Modelo para novas traduções	Vazio
Formato de ficheiro	<i>mi18n lang file</i>

Veja também:

mi18n Mozilla and Java properties files, *Propriedades Java*, updating-target-files, *Formata as propriedades do ficheiro Java*, *Limpeza de ficheiros de tradução*

1.10.8 Propriedades GWT

Formato nativo de GWT para traduções.

As propriedades GWT são geralmente usadas como traduções monolíngues.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	src/app/Bundle_*.properties
Ficheiro de idioma base monolíngue	src/app/Bundle.properties
Modelo para novas traduções	Vazio
Formato de ficheiro	<i>Propriedades GWT</i>

Veja também:

GWT localization guide, GWT Internationalization Tutorial, Mozilla and Java properties files, updating-target-files, *Formata as propriedades do ficheiro Java*, *Limpeza de ficheiros de tradução*

1.10.9 Traduções de INI

Novo na versão 4.1.

Formato de aplicação INI para traduções.

As traduções de INI são geralmente usadas como traduções monolíngues.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	language/*. <i>ini</i>
Ficheiro de idioma base monolíngue	language/en. <i>ini</i>
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro INI</i>

Nota: O Weblate só extrai chaves de seções dentro de um ficheiro INI. Caso o seu ficheiro INI não tenha seções, quereria usar *Traduções para Joomla* ou *Propriedades Java*.

Veja também:

INI Files, *Propriedades Java*, *Traduções para Joomla*, *Traduções Inno Setup INI*

1.10.10 Traduções Inno Setup INI

Novo na versão 4.1.

Formato de ficheiro Inno Setup INI para traduções.

As traduções Inno Setup INI são normalmente usadas como traduções monolíngues.

Nota: A única diferença notável para *Traduções de INI* é ter suporte aos espaços reservados %n and %t para quebra de linha e tabulação.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	language/*. <i>isl</i>
Ficheiro de idioma base monolíngue	language/en. <i>isl</i>
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro Inno Setup INI</i>

Nota: Apenas ficheiros Unicode (*.isl*) são suportados atualmente, a variante ANSI (*.isl*) não é suportada atualmente.

Veja também:

INI Files, *Traduções para Joomla*, *Traduções de INI*

1.10.11 Traduções para Joomla

Novo na versão 2.12.

Formato nativo de Joomla para traduções.

As traduções de Joomla são geralmente usadas como traduções monolíngues.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	language/*/ <i>com_foobar.ini</i>
Ficheiro de idioma base monolíngue	language/en-GB/ <i>com_foobar.ini</i>
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro de idioma Joomla</i>

Veja também:

Especificação de ficheiros de idiomas Joomla, Mozilla and Java properties files, *Traduções de INI*, *Traduções Inno Setup INI*

1.10.12 .ts de Qt Linguist

Formato de tradução usado em aplicações baseados em Qt.

Os ficheiros Qt Linguist são usados como traduções bilíngues e monolíngues.

<i>Configuração de componente</i> típica do Weblate ao usar como bilíngue	
Máscara de ficheiro	<code>i18n/app.*.ts</code>
Ficheiro de idioma base monolingue	<i>Vazio</i>
Modelo para novas traduções	<code>i18n/app.de.ts</code>
Formato de ficheiro	<i>Ficheiro de tradução Qt Linguist</i>

<i>Configuração de componente</i> típica do Weblate ao usar como monolíngue	
Máscara de ficheiro	<code>i18n/app.*.ts</code>
Ficheiro de idioma base monolingue	<code>i18n/app.en.ts</code>
Modelo para novas traduções	<code>i18n/app.en.ts</code>
Formato de ficheiro	<i>Ficheiro de tradução Qt Linguist</i>

Veja também:

Manual do Qt Linguist, Qt .ts, *Formatos bilíngues e monolíngues*

1.10.13 Recurso de cadeias de Android

Formato de ficheiro específico do Android para traduzir aplicações.

Os recursos de cadeias do Android são monolíngues, o *Ficheiro de idioma base monolingue* é armazenado num local diferente dos outros `res/values/strings.xml`.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	<code>res/values-*/strings.xml</code>
Ficheiro de idioma base monolingue	<code>res/values/strings.xml</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Recurso de cadeias de Android</i>

Veja também:

Documentação de recursos de cadeias de Android, Android string resources

Nota: As estruturas de `string-array` do Android não são atualmente suportadas. Para contornar isso, pode quebrar os seus vetores de cadeias:

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

se torna:

```
<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>
```

O *string-array* que aponta para os elementos *string* deve ser armazenado num ficheiro diferente e não ser disponibilizado para tradução.

Este script pode ajudar a pré-processar os seus ficheiros strings.xml e traduções existentes: <https://gist.github.com/paour/11291062>

1.10.14 Cadeias de Apple iOS

Formato de ficheiro específico da Apple para traduzir aplicações, utilizado tanto para traduções de aplicações de iOS quanto de iPhone/iPad.

As cadeias de Apple iOS geralmente são usadas como traduções bilingues.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	Resources/*.lproj/Localizable.strings
Ficheiro de idioma base monolingue	Resources/en.lproj/Localizable.strings ou Resources/Base.lproj/Localizable.strings
Modelo para novas traduções	Vazio
Formato de ficheiro	<i>Cadeias do iOS (UTF-8)</i>

Veja também:

Documentação de «ficheiros de cadeias» da Apple, Mac OSX strings

1.10.15 Cadeias de PHP

As traduções do PHP são geralmente monolingues, por isso é recomendável especificar um ficheiro base com (o que é mais frequentemente os) cadeias em inglês.

Exemplo de ficheiro:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	lang/*/texts.php
Ficheiro de idioma base monolingue	lang/en/texts.php
Modelo para novas traduções	lang/en/texts.php
Formato de ficheiro	<i>Cadeias do PHP</i>

Cadeias de PHP Laravel

Alterado na versão 4.1.

Os ficheiros de localização do PHP de Laravel também são suportados com plurais:

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

Veja também:

PHP, Localização de Laravel

1.10.16 Ficheiro JSON

Novo na versão 2.0.

Alterado na versão 2.16: Desde o Weblate 2.16 e com [translate-toolkit](#) pelo menos 2.2.4, os ficheiros JSON de estrutura aninhada também são suportados.

Alterado na versão 4.3: A estrutura de ficheiros JSON é devidamente preservada mesmo para situações complexas que foram quebradas em versões anteriores.

O formato JSON é usado principalmente para traduzir aplicações implementadas em JavaScript.

O Weblate atualmente suporta várias variantes das traduções JSON:

- Ficheiro simples de chaves/valores, usados por exemplo por *vue-i18n* ou *react-intl*.
- Ficheiro com chaves aninhadas.
- *Ficheiros i18next do JSON*
- *Ficheiros JSON go-i18n*
- *JSON WebExtension*
- *Ficheiro ARB*

As traduções JSON são geralmente monolíngues, por isso é recomendável especificar um ficheiro base com (o que é mais frequentemente os) cadeias em inglês.

Exemplo de ficheiro:

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

Os ficheiros aninhados também são suportados (veja acima para os requisitos), tal ficheiro pode parecer:

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

Dica: O *ficheiro JSON* e *Ficheiro de estrutura JSON aninhada* podem ambos lidar com o mesmo tipo de ficheiros. Ambos preservam a estrutura JSON existente ao traduzir.

A única diferença entre eles é ao adicionar textos ao usar Weblate. O formato de estrutura aninhada analisa a tecla recém-adicionada e insere a cadeia na estrutura de correspondência. Por exemplo, a tecla `app.name` está inserida como:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	<code>langs/translation-*.json</code>
Ficheiro de idioma base monolingue	<code>langs/translation-en.json</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro de estrutura JSON aninhada</i>

Veja também:

[JSON](#), [updating-target-files](#), [Personalizar a saída JSON](#), [Limpeza de ficheiros de tradução](#),

1.10.17 Ficheiros i18next do JSON

Alterado na versão 2.17: Desde o Weblate 2.17 e com [translate-toolkit](#) pelo menos 2.2.5, os ficheiros I18next do JSON com plurais também são suportados.

i18next é uma estrutura de internacionalização escrita dentro e para JavaScript. O Weblate suporta os ficheiros de localização dele com recursos como plurais.

As traduções *i18next* são monolíngues, por isso é recomendável especificar um ficheiro base com (o que é mais frequentemente os) cadeias em inglês.

Nota: Weblate suporta o formato v3 de I18next do JSON. As variantes V2 e V1 são na maioria compatíveis, com exceção de como os plurais são manuseados.

Exemplo de ficheiro:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	<code>langs/*.json</code>
Ficheiro de idioma base monolingue	<code>langs/en.json</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro JSON i18next</i>

Veja também:

[JSON](#), [i18next JSON Format](#), [updating-target-files](#), [Personalizar a saída JSON](#), [Limpeza de ficheiros de tradução](#)

1.10.18 Ficheiros JSON go-i18n

Novo na versão 4.1.

As traduções go-i18n são monolíngues, por isso é recomendável especificar um ficheiro base com (o que é mais frequentemente os) cadeias em inglês.

Nota: Weblate tem suporte ao formato go-i18n JSON v1, para formatos JSON simples, use *Ficheiro JSON*. O formato v2 com hash não é suportado no momento.

<i>Configuração de componente típica do Weblate</i>	
Máscara de ficheiro	langs/*.json
Ficheiro de idioma base monolíngue	langs/en.json
Modelo para novas traduções	Vazio
Formato de ficheiro	ficheiro JSON go-i18n

Veja também:

[JSON](#), [go-i18n](#), [updating-target-files](#), [Personalizar a saída JSON](#), [Limpeza de ficheiros de tradução](#),

1.10.19 Ficheiro ARB

Novo na versão 4.1.

As traduções ARB são monolíngues, por isso é recomendável especificar um ficheiro base com (o que é mais frequentemente os) cadeias em inglês.

<i>Configuração de componente típica do Weblate</i>	
Máscara de ficheiro	lib/l10n/intl_*.arb
Ficheiro de idioma base monolíngue	lib/l10n/intl_en.arb
Modelo para novas traduções	Vazio
Formato de ficheiro	Ficheiro ARB

Veja também:

[JSON](#), [Application Resource Bundle Specification](#), [Internationalizing Flutter apps](#), [updating-target-files](#), [Personalizar a saída JSON](#), [Limpeza de ficheiros de tradução](#)

1.10.20 JSON WebExtension

Novo na versão 2.16: Isto é suportado desde Weblate 2.16 e com o [translate-toolkit](#) pelo menos 2.2.4.

Formato de ficheiro usado ao traduzir extensões para Mozilla Firefox ou Google Chromium.

Nota: Embora este formato se chame JSON, a especificação dele permite incluir comentários, que não fazem parte da especificação JSON. O Weblate atualmente não suporta ficheiro com comentários.

Exemplo de ficheiro:

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

(continues on next page)

(continuação da página anterior)

```

        "content": "$1",
        "example": "https://developer.mozilla.org"
    }
},
"orangutan": {
    "message": "",
    "description": "Description"
},
"try": {
    "message": "",
    "description": "Description"
},
"thanks": {
    "message": "",
    "description": "Description"
}
}

```

Configuração de componente típica do Weblate

Máscara de ficheiro	_locales/*/messages.json
Ficheiro de idioma base monolingue	_locales/en/messages.json
Modelo para novas traduções	Vazio
Formato de ficheiro	Ficheiro JSON WebExtension

Veja também:

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

1.10.21 Ficheiros de recursos .XML

Novo na versão 2.3.

Um ficheiro de recurso .XML (.resx) emprega um formato de ficheiro XML monolíngue usado nas aplicações Microsoft .NET. É intercambiável com .resw, quando usa sintaxe idêntica ao .resx.

Configuração de componente típica do Weblate

Máscara de ficheiro	Resources/Language.*.resx
Ficheiro de idioma base monolingue	Resources/Language.resx
Modelo para novas traduções	Vazio
Formato de ficheiro	Ficheiro de recursos .NET

Veja também:

.NET Resource files (.resx), [updating-target-files](#), [Limpeza de ficheiros de tradução](#)

1.10.22 Ficheiros CSV

Novo na versão 2.4.

Os ficheiros CSV podem conter uma simples lista de origem e tradução. O Weblate suporta os seguintes ficheiros:

- Ficheiros com campos de definição de cabeçalho (`location`, `source`, `target`, `ID`, `fuzzy`, `context`, `translator_comments`, `developer_comments`). Esta é a abordagem recomendada, pois é a menos sujeita a erros. Escolha *Ficheiro CSV* como um formato de ficheiro.
- Ficheiros com dois campos – fonte e tradução (nesta ordem). Escolha *Ficheiro CSV simples* como um formato de ficheiro.
- 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.
- Lembre-se de definir o *Ficheiro de idioma base monolíngue* quando os seus ficheiros são monolíngues (veja *Formatos bilíngues e monolíngues*).

Aviso: O formato CSV detecta automaticamente o dialeto do ficheiro CSV. Em alguns casos, a detecção automática pode falhar e terá resultados mistos. Isso é especialmente verdadeiro para ficheiros CSV com linhas novas nos valores. Como solução alternativa, recomenda-se omitir caracteres entre aspas.

Exemplo de ficheiro:

Thank you for using Weblate.,Děkuji za použití Weblate.

<i>Configuração de componente</i> típica do Weblate para CSV bilíngue	
Máscara de ficheiro	<code>locale/*.csv</code>
Ficheiro de idioma base monolíngue	<i>Vazio</i>
Modelo para novas traduções	<code>locale/en.csv</code>
Formato de ficheiro	<i>Ficheiro CSV</i>

<i>Configuração de componente</i> típica do Weblate para CSV monolíngue	
Máscara de ficheiro	<code>locale/*.csv</code>
Ficheiro de idioma base monolíngue	<code>locale/en.csv</code>
Modelo para novas traduções	<code>locale/en.csv</code>
Formato de ficheiro	<i>Ficheiro CSV simples</i>

Veja também:

CSV

1.10.23 Ficheiros YAML

Novo na versão 2.9.

Os ficheiros YAML simples com chaves de cadeias e valores. Weblate também extrai cadeias de listas ou dicionários.

Exemplo de um ficheiro YAML:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	translations/messages.*.yaml
Ficheiro de idioma base monolíngue	translations/messages.en.yaml
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro YAML</i>

Veja também:

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

1.10.24 Ficheiros Ruby YAML

Novo na versão 2.9.

Ficheiros YAML de i18n do Ruby com idioma como nó raiz.

Exemplo de ficheiro YAML de i18n do Ruby:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	translations/messages.*.yaml
Ficheiro de idioma base monolíngue	translations/messages.en.yaml
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro Ruby YAML</i>

Veja também:

[YAML](#), [Ficheiros YAML](#)

1.10.25 Ficheiros DTD

Novo na versão 2.18.

Exemplo de ficheiro DTD:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	locale/*.dtd
Ficheiro de idioma base monolíngue	locale/en.dtd
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro DTD</i>

Veja também:

[Mozilla DTD format](#)

1.10.26 Ficheiro XML simples

Novo na versão 3.9.

Exemplo de um ficheiro XML simples:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	locale/*.xml
Ficheiro de idioma base monolíngue	locale/en.xml
Modelo para novas traduções	Vazio
Formato de ficheiro	<i>Ficheiro XML simples</i>

Veja também:

[Flat XML](#)

1.10.27 Ficheiros RC do Windows

Alterado na versão 4.1: O suporte para ficheiros RC do Windows foi reescrito.

Nota: O suporte para este formato está atualmente em beta, feedback de testes é bem-vindo.

Exemplo de ficheiro RC do Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

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

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	lang/*.rc
Ficheiro de idioma base monolíngue	lang/en-US.rc
Modelo para novas traduções	lang/en-US.rc
Formato de ficheiro	<i>Ficheiro RC</i>

Veja também:

[Windows RC files](#)

1.10.28 Ficheiros de metadados da App Store

Novo na versão 3.5.

Metadados usados para publicar aplicações em várias lojas de aplicações podem ser traduzidos. Atualmente, as seguintes ferramentas são compatíveis:

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

Os metadados consistem em vários ficheiros de texto, que o Weblate apresentará como cadeias separados para traduzir.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	<code>fastlane/android/metadata/*</code>
Ficheiro de idioma base monolingue	<code>fastlane/android/metadata/en-US</code>
Modelo para novas traduções	<code>fastlane/android/metadata/en-US</code>
Formato de ficheiro	<i>Ficheiros de metadados de loja de aplicações</i>

Dica: Caso não queira traduzir certas cadeias (por exemplo, changelogs), marque-as como somente leitura (veja *Personalizando comportamento a usar marcadores*). Isto pode ser automatizado por *Edição em massa*.

1.10.29 Ficheiros de legenda

Novo na versão 3.7.

Weblate pode traduzir vários ficheiros de legenda:

- Ficheiros de legenda SubRip (*.srt)
- Ficheiro de legenda MicroDVD (*.sub)
- Ficheiro de legenda Advanced Substation Alpha (*.ass)
- Ficheiro de legenda Substation Alpha (*.ssa)

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	<code>path/*.srt</code>
Ficheiro de idioma base monolingue	<code>path/en.srt</code>
Modelo para novas traduções	<code>path/en.srt</code>
Formato de ficheiro	<i>Ficheiro de legenda SubRip</i>

Veja também:

[Subtitles](#)

1.10.30 Excel Open XML

Novo na versão 3.2.

Os ficheiros Excel Open XML (.xlsx) podem ser importados e exportados.

Ao enviar ficheiros XLSX para a tradução, esteja ciente que apenas a planilha ativa é considerada e deve haver pelo menos uma coluna chamada `source` (que contém a cadeia fonte) e uma coluna chamada `target` (que contém a tradução). Além disso, deve haver a coluna chamada `context` (que contém o caminho de contexto da cadeia de tradução). Se usar a descarrega XLSX para exportar as traduções numa pasta de trabalho do Excel, já terá um ficheiro com o formato de ficheiro correto.

1.10.31 Ficheiro HTML

Novo na versão 4.1.

Nota: O suporte para este formato está atualmente em beta, feedback de testes é bem-vindo.

O conteúdo traduzível é extraído dos ficheiros HTML e oferecido para a tradução.

Veja também:

[HTML](#)

1.10.32 Ficheiros de texto

Novo na versão 4.6.

Nota: O suporte para este formato está atualmente em beta, feedback de testes é bem-vindo.

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

Há três sabores desse formato:

- Ficheiros de texto simples
- Ficheiros de texto DokuWiki
- Ficheiro de texto MediaWiki

Veja também:

[Simple Text Documents](#)

1.10.33 Formato OpenDocument

Novo na versão 4.1.

Nota: O suporte para este formato está atualmente em beta, feedback de testes é bem-vindo.

O conteúdo traduzível é extraído dos ficheiros OpenDocument e oferecido para a tradução.

Veja também:

[OpenDocument Format](#)

1.10.34 Formato IDML

Novo na versão 4.1.

Nota: O suporte para este formato está atualmente em beta, feedback de testes é bem-vindo.

O conteúdo traduzível é extraído dos ficheiros Adobe InDesign Markup Language e oferecido para a tradução.

1.10.35 Formato TermBase eXchange

Novo na versão 4.5.

TBX é um formato XML para a troca de dados de terminologia.

<i>Configuração de componente</i> típica do Weblate	
Máscara de ficheiro	<code>tbx/*.*tbx</code>
Ficheiro de idioma base monolingue	<i>Vazio</i>
Modelo para novas traduções	<i>Vazio</i>
Formato de ficheiro	<i>Ficheiro TermBase eXchange</i>

Veja também:

TBX na Wikipédia, TBX, *Glossário*

1.10.36 Supporting other formats

A maioria dos formatos suportados pelo [translate-toolkit](#) que tem suporte a serialização pode ser facilmente suportado, mas eles não (ainda) receberam nenhum teste. Na maioria dos casos, alguma camada fina é necessária no Weblate para ocultar diferenças no comportamento de diferentes armazenamentos do [translate-toolkit](#).

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

Veja também:

[Translation Related File Formats](#)

1.11 Integração de controlo de versões

Weblate atualmente tem suporte a *Git* (com suporte estendido a *GitHub*, *Gerrit* e *Subversion*) e *Mercurial* como back-ends de controle de versão.

1.11.1 Acessando repositórios

O repositório VCS que deseja usar tem que ser acessível ao Weblate. Com um repositório disponível publicamente, só precisa inserir a URL correta (por exemplo `https://github.com/WeblateOrg/weblate.git`), mas para repositórios privados ou para URLs de push a configuração é mais complexa e requer autenticação.

Acessando repositórios do Hosted Weblate

Para Hosted Weblate há um utilizador dedicado para fazer push registado no GitHub, Bitbucket, Codeberg e GitLab (com o nome de utilizador *weblate*, e-mail `hosted@weblate.org` e chamado *Weblate push user*). Precisa adicionar esse utilizador como colaborador e dar a permissão apropriada ao seu repositório (somente leitura está bom para clonagem, escrita é necessária para fazer push). Dependendo do serviço e das configurações da sua organização, isso acontece imediatamente, ou requer confirmação do lado do Weblate.

O utilizador *weblate* no GitHub aceita convites automaticamente dentro de cinco minutos. O processamento manual pode ser necessário nos outros serviços, por isso, por favor, seja paciente.

Uma vez adicionado o utilizador *weblate*, pode configurar o *Repositório do código-fonte* e a *URL de submissão do repositório* a utilizar o protocolo SSH (por exemplo, `git@github.com:WeblateOrg/weblate.git`).

Repositórios SSH

O método mais usado para acessar repositórios privados é baseado no SSH. Autorize a chave pública SSH do Weblate (veja *Chave SSH do Weblate*) para acessar o repositório upstream desta forma.

Aviso: No GitHub, cada chave só pode ser utilizada uma vez, veja `vcs-repos-github` e `:ref:~hosted-push`.

Weblate também armazena a impressão digital da chave do host na primeira conexão e não se conecta ao host caso ele seja alterado posteriormente (veja *Verificando chaves SSH do host*).

Caso o ajuste seja necessário, faça-o a partir da interface de administração Weblate:

The screenshot shows the 'SSH keys' section of the Weblate administration interface. At the top, there's a navigation bar with 'Weblate' logo and links to Dashboard, Projects, Languages, and Checks. Below this is a sub-header 'Manage / SSH keys' and a row of tabs: Weblate status, Backups, Translation memory, Performance report, SSH keys (active), Alerts, Repositories, Users, and Appearance. Under the 'SSH keys' tab, there are two sections: 'Public SSH key' and 'Known host keys'. The 'Public SSH key' section shows a text area with an SSH public key for 'ssh-rsa' and a 'Download private key' button. The 'Known host keys' section is a table with columns for Hostname, Key type, and Fingerprint, showing an entry for 'github.com'. Below this is an 'Add host key' section with input fields for Hostname and Port, and a 'Submit' button.

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8

Chave SSH do Weblate

A chave pública do Weblate está visível para todos os utilizadores que navegam na página *Sobre*.

Os administradores podem gerar ou exibir a chave pública usada atualmente pelo Weblate na conexão (a partir de *Chaves SSH*) na página inicial da interface administrativa.

Nota: A chave SSH privada correspondente não pode ter uma palavra-passe no momento, por isso certifique-se de que ela está bem protegida.

Dica: Faça um backup da chave SSH privada gerada do Weblate.

Verificando chaves SSH do host

O Weblate armazena automaticamente as chaves SSH do host no primeiro acesso e lembra-se delas para uso posterior.

Caso queira verificar a impressão digital da chave antes de se conectar ao repositório, adicione as chaves SSH dos servidores que vai acessar em *Adicionar chave de host*, a partir da mesma secção da interface de administração. Digite o nome do host que vai acessar (por exemplo, `gitlab.com`) e pressione *Enviar*. Verifique se a sua impressão digital corresponde ao servidor que adicionou.

As chaves adicionadas com impressões digitais são mostradas na mensagem de confirmação:

Added host key for github.com with fingerprint nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8 (ssh-rsa), please verify that it is correct.

[Weblate status](#)
[Backups](#)
[Translation memory](#)
[Performance report](#)
[SSH keys](#)
[Alerts](#)
[Repositories](#)
[Users](#)
[Appearance](#)

[Tools](#)
[Billing](#)

Public SSH key

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

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCR5HIujkPOvcqhouFxcnFboxi+dKKS9w3FOsjYjr5ZjaZ2tmTX7gRRmVa9xBldFYIRxq/t5n+ImmjgvjLRMteYDKL6i9Ndd5aMsN
Weblate
```

[Download private key](#)

Known host keys

Hostname	Key type	Fingerprint
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname
Port

[Submit](#)

Repositórios do GitHub

O acesso via SSH é possível (veja [Repositórios SSH](#)), mas caso precise acessar mais de um repositório, atingirá uma limitação do GitHub no uso permitido da chave SSH (já que cada chave pode ser usada apenas uma vez).

Caso o *Ramo do push* não seja definido, é criado um fork do projeto e feito um push das alterações através do fork. Caso seja definido, os pushes são feitos para o repositório upstream e para o ramo escolhido.

Para implantações menores, use autenticação HTTPS com um token de acesso pessoal e a sua conta no GitHub, veja [Criando um token de acesso para uso em linha de comando](#).

Para configurações maiores, geralmente é melhor criar um utilizador dedicado para o Weblate, atribuir-lhe a chave SSH pública gerada no Weblate (ver [Chave SSH do Weblate](#)) e concedê-lo acesso a todos os repositórios que deseja traduzir. Essa abordagem também é usada para o Hosted Weblate, há utilizador dedicado *weblate* para isso.

Veja também:

[Acessando repositórios do Hosted Weblate](#)

URLs internas do Weblate

Compartilhe uma configuração de repositório entre diferentes componentes, a fazer referência à sua colocação como `weblate://projeto/componente` em outros componentes (vinculados). Desta forma, os componentes vinculados utilizam a configuração do repositório VCS do componente principal (referenciado).

Aviso: A remoção do componente principal também remove componentes vinculados.

O Weblate ajusta automaticamente a URL do repositório ao criar um componente se encontrar um componente com uma configuração de repositório correspondente. Pode anular isso na última etapa da configuração do componente.

Motivos para usar isso:

- Economiza espaço em disco no servidor, o repositório é armazenado apenas uma vez.
- Torna as atualizações mais rápidas, apenas um repositório é atualizado.
- Há apenas um repositório exportado com traduções do Weblate (ver [Exportador git](#)).
- Algumas extensões podem operar em vários componentes a compartilhar um repositório; por exemplo, [Squash de commits git](#).

Repositórios HTTPS

Para acessar repositórios HTTPS protegidos, inclua o nome de utilizador e a palavra-passe na URL. Não se preocupe, o Weblate irá remover essas informações quando a URL for mostrada aos utilizadores (se mesmo permitido ver a URL do repositório).

Por exemplo, a URL do GitHub com autenticação adicionada pode parecer: `https://usuario:seu_token_de_acesso@github.com/WeblateOrg/weblate.git`.

Nota: Se o seu nome de utilizador ou palavra-passe contiver caracteres especiais, eles devem ser codificados para URL; por exemplo, `https://usuario%40example.com:%24senha%23@bitbucket.org/....`

Usando proxy

Se precisar acessar repositórios VCS por HTTP/HTTPS a usar um servidor proxy, configure o VCS para usá-lo.

Isto pode ser feito a utilizar as variáveis de ambiente `http_proxy`, `https_proxy` e `all_proxy` (como descrito na documentação do [cURL](#)) ou a aplicar-a na configuração do VCS, por exemplo:

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

Nota: A configuração do proxy precisa ser feita com o utilizador a executar Weblate (veja também *Permissões do sistema de ficheiros*) e com `HOME=$DATA_DIR/home` (veja *DATA_DIR*), caso contrário o Git executado pelo Weblate não o utilizará.

Veja também:

A página [man do cURL](#), [Documentação do git config](#)

1.11.2 Git

Veja também:

Consulte *Acessando repositórios* para obter informações sobre como acessar diferentes tipos de repositórios.

Git com push forçado

Ele se comporta exatamente como o próprio Git, a única diferença é que ele sempre força pushes. Isso se destina apenas no caso de usar um repositório separado para traduções.

Aviso: Use com cautela, pois isso facilmente leva a commits perdidos no seu repositório upstream.

Personalizando a configuração do Git

Weblate invoca todos os comandos VCS com `HOME=$DATA_DIR/home` (veja `:set:DATA_DIR`), portanto a edição da configuração do utilizador precisa ser feita em `DATA_DIR/home/.git`.

Auxiliares de remotos do Git

Também pode usar os *auxiliares de remotos* do Git para ter suporte adicionalmente a outros sistemas de controle de versão, mas esteja preparado para depurar problemas que isso pode levar.

Neste momento, os auxiliares de Bazaar e Mercurial estão disponíveis em repositórios separados no GitHub: [git-remote-hg](#) e [git-remote-bzr](#). Baixe-os manualmente e ponha em algum lugar no seu caminho de pesquisa (por exemplo, `:file:~/bin`). Certifique-se de ter os sistemas de controle de versão correspondentes instalados.

Uma vez instalados, esses controles podem ser usados para especificar um repositório no Weblate.

Para clonar o projeto `gnuhello` do Launchpad a usar Bazaar:

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

Para o repositório `hello` de `selenic.com` a usar Mercurial:

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

Aviso: O inconveniente de usar auxiliares de remotos Git é, por exemplo, com o Mercurial, o auxiliar de remoto às vezes cria uma nova dica ao fazer push das mudanças de volta.

1.11.3 GitHub

Novo na versão 2.3.

Isto adiciona uma camada fina sobre o *Git* a utilizar a [API do GitHub](#) para permitir fazer push de alterações de tradução como pull requests, ao invés de fazer push diretamente para o repositório.

Git faz push das alterações diretamente para um repositório, enquanto *GitHub* cria pull requests. Este último não é necessário para apenas acessar repositórios Git.

Veja também:

Fazendo push das alterações do Weblate

Fazendo push de alterações para o GitHub como pull requests

Se não quiser fazer push das traduções para um repositório do GitHub, elas podem ser enviadas como uma ou várias pull requests.

Precisa configurar credenciais de API para fazer isso funcionar.

Veja também:

GITHUB_USERNAME, GITHUB_TOKEN, GITHUB_CREDENTIALS

1.11.4 GitLab

Novo na versão 3.9.

Isto apenas adiciona uma camada fina sobre o *Git* a usar a [API do GitLab](#) para permitir fazer push de alterações de tradução como merge requests, ao invés de fazer push diretamente para o repositório.

Não há necessidade de usá-lo para acessar repositórios de Git, o *Git* comum funciona da mesma forma, a ser a única diferença como o push para um repositório é manipulado. Com *Git*, o push das alterações é feito diretamente para o repositório, enquanto *GitLab* cria merge request.

Veja também:

Fazendo push das alterações do Weblate

Fazendo push de alterações para o GitLab como merge requests

Se não quiser fazer push das traduções para um repositório do GitLab, elas podem ser enviadas como uma ou várias merge requests.

Precisa configurar credenciais de API para fazer isso funcionar.

Veja também:

GITLAB_USERNAME, GITLAB_TOKEN, GITLAB_CREDENTIALS

1.11.5 Pagure

Novo na versão 4.3.2.

Isto apenas adiciona uma camada fina sobre o [Git](#) a usar a [API do Pagure](#) para permitir fazer push de alterações de tradução como merge requests, ao invés de fazer push diretamente para o repositório.

Não há necessidade de usá-lo para acessar repositórios de Git, o [Git](#) comum funciona da mesma forma, a ser a única diferença como o push para um repositório é manipulado. Com [Git](#), o push das alterações é feito diretamente para o repositório, enquanto [Pagure](#) cria merge request.

Veja também:

Fazendo push das alterações do Weblate

Enviar alterações para o Pagure como solicitações de mesclagem

Se não quiser fazer push das traduções para um repositório do Pagure, elas podem ser enviadas como uma ou várias merge requests.

Precisa configurar credenciais de API para fazer isso funcionar.

Veja também:

`PAGURE_USERNAME`, `PAGURE_TOKEN`, `PAGURE_CREDENTIALS`

1.11.6 Gerrit

Novo na versão 2.2.

Adiciona uma camada fina sobre o [Git](#) a usar a ferramenta [git-review](#) para permitir fazer push de alterações de tradução como review requests do Gerrit, ao invés de fazer push diretamente para o repositório.

A documentação Gerrit tem os detalhes sobre a configuração necessária para configurar tais repositórios.

1.11.7 Mercurial

Novo na versão 2.1.

Mercurial é outro VCS que pode usar diretamente no Weblate.

Nota: Ele deve funcionar com qualquer versão Mercurial, mas às vezes há alterações incompatíveis na interface de linha de comando que quebra a integração Weblate.

Veja também:

Consulte [Acessando repositórios](#) para obter informações sobre como acessar diferentes tipos de repositórios.

1.11.8 Subversion

Novo na versão 2.8.

O Weblate usa [git-svn](#) para interagir com repositórios [subversion](#). É um script Perl que permite que o subversion seja usado por um cliente Git, a permitir que os utilizadores mantenham um clone completo do repositório interno e façam commit localmente.

Nota: O Weblate tenta detetar o layout do repositório Subversion automaticamente – ele tem suporte a URLs diretas para repositórios com layout padrão (branches/, tags/ e trunk/). Mais informações sobre isso podem ser

encontradas na [documentação do git-svn](#). Se o repositório não tiver um layout padrão e encontrar erros, tente incluir o nome do ramo na URL do repositório e deixar a ramo vazia.

Alterado na versão 2.19: Antes disso, apenas repositórios a usar o layout padrão eram suportados.

Credenciais de Subversion

Weblate espera que tenha aceito o certificado com antecedência (e as suas credenciais, se necessário). Ele procurará inseri-las no diretório `:set:DATA_DIR`. Aceite o certificado a utilizar *svn* uma vez com a variável de ambiente `$HOME` definida como `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
```

Veja também:

`DATA_DIR`

1.11.9 Ficheiros locais

Novo na versão 3.8.

O Weblate também pode operar sem um VCS remoto. As traduções iniciais são importadas a carrega-las. Mais tarde, pode substituir ficheiros individuais a enviar ficheiros ou a adicionar cadeias de tradução diretamente do Weblate (atualmente disponível apenas para traduções monolíngues).

No fundo, o Weblate cria um repositório de Git para si e todas as alterações são rastreadas. Caso decida mais tarde usar um VCS para armazenar as traduções, já tem um repositório dentro do Weblate pode basear na sua integração.

1.12 API REST do Weblate

Novo na versão 2.6: A API REST está disponível desde o Weblate 2.6.

A API é acessível na URL `/api/` e é baseada em [framework Django REST](#). Pode utilizá-la diretamente ou por *Cliente Weblate*.

1.12.1 Autenticação e parâmetros genéricos

A API do projeto público está disponível sem autenticação, embora as solicitações não autenticadas sejam fortemente limitadas (por padrão a 100 solicitações por dia), por isso é recomendado usar autenticação. A autenticação usa um token, que pode obter no seu perfil. Use-o no cabeçalho `Authorization`:

ANY /

O comportamento de solicitação genérico para a API, os cabeçalhos, códigos de estado e parâmetros aqui aplicam-se a todos os pontos finais também.

Parâmetros de consulta

- **format** – Formato de resposta (substitui `Accept`). Os valores possíveis dependem da configuração do quadro REST, por padrão `json` e `api` são suportados. Este último fornece uma interface de navegador web para API.

Cabeçalhos de solicitação

- `Accept` – o tipo de conteúdo de resposta depende do cabeçalho `Accept`
- `Authorization` – optional token to authenticate as `Authorization: Token YOUR-TOKEN`

Cabeçalhos de resposta

- **Content-Type** – isso depende de cabeçalho **Accept** da solicitação
- **Allow** – lista de métodos HTTP permitidos no objeto

Objeto JSON de resposta

- **detail** (*string*) – descrição detalhada do resultado (para códigos de estado HTTP diferentes de 200 OK)
- **count** (*int*) – contagem total de itens para listas de objetos
- **next** (*string*) – URL da próxima página para listas de objetos
- **previous** (*string*) – URL da página anterior para listas de objetos
- **results** (*array*) – resultados para listas de objetos
- **url** (*string*) – URL para acessar esse recurso a usar API
- **web_url** (*string*) – URL para acessar esse recurso a usar navegador web

Códigos de estado

- 200 OK – quando o pedido foi resolvido corretamente
- 201 Created – when a new object was created successfully
- 204 No Content – when an object was deleted successfully
- 400 Bad Request – quando faltam os parâmetros do formulário
- 403 Forbidden – quando o acesso é negado
- 429 Too Many Requests – quando a limitação está em vigor

Exemplos de autenticação

Exemplo de solicitação:

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

Exemplo de resposta:

```
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/"
}
```

Exemplo de CURL:

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

Exemplos de passagem de parâmetros

Para o método **POST** os parâmetros podem ser especificados como envio de formulário (*application/x-www-form-urlencoded*) ou como JSON (*application/json*).

Exemplo de solicitação de formulário:

```
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
```

Exemplo de solicitação JSON:

```
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"}
```

Exemplo de CURL:

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

Exemplo de CURL JSON:

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

Limitação de taxa da API

As solicitações de API são limitadas; a configuração padrão limita-a a 100 solicitações por dia para utilizadores anônimos e 5000 solicitações por hora para utilizadores autenticados.

A limitação da taxa pode ser ajustada no `settings.py`. Consulte [Throttling na documentação do framework Django REST](#) para obter mais detalhes sobre como configurá-la.

O estado de limitação de taxa é relatado nos seguintes cabeçalhos:

X-RateLimit-Limit	Limite de limitação de taxa de solicitações para realizar
X-RateLimit-Remaining	Limite restante de solicitações
X-RateLimit-Reset	Quantidade de segundos até que a janela de «ratelimit» seja reiniciada

Alterado na versão 4.1: Adicionados cabeçalhos de estado de limitação de taxa.

Veja também:

Limitação de taxa, Limitação de taxa

1.12.2 Ponto de entrada da API

GET /api/

O ponto de entrada raiz da API.

Exemplo de solicitação:

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

Exemplo de resposta:

```
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 Utilizadores

Novo na versão 4.0.

GET /api/users/

Retorna uma lista de utilizadores se tiver permissões para ver utilizadores geridos. Se não, então pode ver apenas os seus próprios detalhes.

Veja também:

Os atributos do objeto dos utilizadores estão documentados em `GET /api/users/(str:username)/`.

POST /api/users/

Cria um novo utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador
- **full_name** (*string*) – Nome completo do utilizador
- **email** (*string*) – E-mail do utilizador
- **is_superuser** (*boolean*) – O utilizador é superutilizador? (opcional)
- **is_active** (*boolean*) – O utilizador está ativo? (opcional)

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

Retorna informações sobre os utilizadores.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

Objeto JSON de resposta

- **username** (*string*) – nome de utilizador de um utilizador
- **full_name** (*string*) – nome completo de um utilizador
- **email** (*string*) – email de um utilizador
- **is_superuser** (*boolean*) – se o utilizador é um superutilizador
- **is_active** (*boolean*) – se o utilizador está ativo
- **date_joined** (*string*) – data de criação do utilizador
- **groups** (*array*) – ligações para grupos associados; veja `GET /api/groups/(int:id)/`

Exemplo de dados JSON:

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

PUT /api/users/ (str: username) /
Altera os parâmetros do utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

Objeto JSON de resposta

- **username** (*string*) – nome de utilizador de um utilizador
- **full_name** (*string*) – nome completo de um utilizador
- **email** (*string*) – email de um utilizador
- **is_superuser** (*boolean*) – se o utilizador é um superutilizador
- **is_active** (*boolean*) – se o utilizador está ativo
- **date_joined** (*string*) – data de criação do utilizador

PATCH /api/users/ (str: username) /
Altera os parâmetros do utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

Objeto JSON de resposta

- **username** (*string*) – nome de utilizador de um utilizador
- **full_name** (*string*) – nome completo de um utilizador

- **email** (*string*) – email de um utilizador
- **is_superuser** (*boolean*) – se o utilizador é um superutilizador
- **is_active** (*boolean*) – se o utilizador está ativo
- **date_joined** (*string*) – data de criação do utilizador

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

Apaga todas as informações do utilizador e marca o utilizador como inativo.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

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

Associa grupos a um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

Parâmetros de formulário

- **string group_id** – O ID de grupo único

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

Lista as estatísticas de um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

Objeto JSON de resposta

- **translated** (*int*) – Quantidade de traduções por utilizador
- **suggested** (*int*) – Quantidade de sugestões por utilizador
- **uploaded** (*int*) – Quantidade de envios por utilizador
- **commented** (*int*) – Quantidade de comentários por utilizador
- **languages** (*int*) – Quantidade de idiomas que o utilizador pode traduzir

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

Lista de subscrições de um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

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

Associar subscrições a um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador

Objeto JSON da solicitação

- **notification** (*string*) – Nome da notificação registada
- **scope** (*int*) – Escopo de notificação a partir das opções disponíveis
- **frequency** (*int*) – Escolha de frequência para notificações

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

int: *subscription_id* / Obtém uma assinatura associada a um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador
- **subscription_id** (*int*) – ID da notificação registada

PUT `/api/users/ (str: username) /notifications/`
int: `subscription_id` / Edita uma assinatura associada a um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador
- **subscription_id** (*int*) – ID da notificação registada

Objeto JSON da solicitação

- **notification** (*string*) – Nome da notificação registada
- **scope** (*int*) – Escopo de notificação a partir das opções disponíveis
- **frequency** (*int*) – Escolha de frequência para notificações

PATCH `/api/users/ (str: username) /notifications/`
int: `subscription_id` / Edita uma assinatura associada a um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador
- **subscription_id** (*int*) – ID da notificação registada

Objeto JSON da solicitação

- **notification** (*string*) – Nome da notificação registada
- **scope** (*int*) – Escopo de notificação a partir das opções disponíveis
- **frequency** (*int*) – Escolha de frequência para notificações

DELETE `/api/users/ (str: username) /notifications/`
int: `subscription_id` / Apaga uma assinatura associada a um utilizador.

Parâmetros

- **username** (*string*) – Nome de utilizador do utilizador
- **subscription_id** – Nome da notificação registada
- **subscription_id** – int

1.12.4 Grupos

Novo na versão 4.0.

GET `/api/groups/`
Retorna uma lista de grupos se tiver permissões para ver grupos geridos. Se não, então pode ver apenas os grupos dos quais o utilizador faz parte.

Veja também:

Os atributos de objeto de grupo estão documentados em `GET /api/groups/(int:id)/`.

POST `/api/groups/`
Cria um novo grupo.

Parâmetros

- **name** (*string*) – Nome do grupo
- **project_selection** (*int*) – Grupo de seleção de projetos a partir de opções dadas
- **language_selection** (*int*) – Grupo de idiomas selecionados a partir de opções dadas

GET `/api/groups/(int: id) /`
Devolve informação sobre o grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

- **name** (*string*) – nome de um grupo
- **project_selection** (*int*) – inteiro correspondente ao grupo de projetos
- **language_selection** (*int*) – inteiro correspondente ao grupo de idiomas
- **roles** (*array*) – ligação para funções associadas; veja `GET /api/roles/(int:id)/`
- **projects** (*array*) – ligação para projetos associados; veja `GET /api/projects/(string:project)/`
- **components** (*array*) – ligação para componentes associados; veja `GET /api/components/(string:project)/(string:component)/`
- **componentlist** (*array*) – ligação para lista de componentes associada; veja `GET /api/component-lists/(str:slug)/`

Exemplo de dados JSON:

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

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

Altera os parâmetros do grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

- **name** (*string*) – nome de um grupo
- **project_selection** (*int*) – inteiro correspondente ao grupo de projetos
- **language_selection** (*int*) – inteiro correspondente ao grupo de idiomas

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

Altera os parâmetros do grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

- **name** (*string*) – nome de um grupo
- **project_selection** (*int*) – inteiro correspondente ao grupo de projetos
- **language_selection** (*int*) – inteiro correspondente ao grupo de idiomas

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

Apaga o grupo.

Parâmetros

- **id** (*int*) – ID do grupo

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

Associar funções a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

- **string role_id** – O ID de função único

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

Associa componentes a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

- **string component_id** – O ID de componente único

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

int: component_id Eliminar componente de um grupo.

Parâmetros

- **id** (*int*) – ID do grupo
- **component_id** (*int*) – O ID de componente único

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

Associa projetos a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

- **string project_id** – O ID do projeto único

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

int: project_id Eliminar projeto de um grupo.

Parâmetros

- **id** (*int*) – ID do grupo
- **project_id** (*int*) – O ID do projeto único

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

Associar línguas a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

- **string language_code** – O código de idioma único

DELETE /api/groups/ (int: id) /languages/
string: *language_code* Eliminar língua de um grupo.

Parâmetros

- **id (int)** – ID do grupo
- **language_code (string)** – O código de idioma único

POST /api/groups/ (int: id) /componentlists/
 Associa listas de componentes a um grupo.

Parâmetros

- **id (int)** – ID do grupo

Parâmetros de formulário

- **string component_list_id** – O ID de lista de componentes único

DELETE /api/groups/ (int: id) /componentlists/
int: *component_list_id* Apaga a lista de componentes de um grupo.

Parâmetros

- **id (int)** – ID do grupo
- **component_list_id (int)** – O ID de lista de componentes único

1.12.5 Funções

GET /api/roles/

Retorna uma lista de todas as funções associadas ao utilizador. Se o utilizador for superutilizador, então a lista de todas as funções existentes será retornada.

Veja também:

Os atributos de objeto de funções estão documentados em *GET /api/roles/ (int:id) /*.

POST /api/roles/

Cria uma nova função.

Parâmetros

- **name (string)** – Nome da função
- **permissions (array)** – Lista de codinomes de permissões

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

Retorna informações sobre uma função.

Parâmetros

- **id (int)** – ID da função

Objeto JSON de resposta

- **name (string)** – Nome da função
- **permissions (array)** – lista de codinomes de permissões

Exemplo de dados JSON:

```
{
  "name": "Access repository",
  "permissions": [
    "vcs.access",
    "vcs.view"
  ]
}
```

(continues on next page)

```
  ],
  "url": "http://example.com/api/roles/1/",
}
```

PUT `/api/roles/(int: id) /`
Altera os parâmetros da função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

- **name** (*string*) – Nome da função
- **permissions** (*array*) – lista de codinomes de permissões

PATCH `/api/roles/(int: id) /`
Altera os parâmetros da função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

- **name** (*string*) – Nome da função
- **permissions** (*array*) – lista de codinomes de permissões

DELETE `/api/roles/(int: id) /`
Apaga a função.

Parâmetros

- **id** (*int*) – ID da função

1.12.6 Idiomas

GET `/api/languages/`
Retorna uma lista de todos os idiomas.

Veja também:

Os atributos do objeto de idioma estão documentados em `GET /api/languages/(string: language) /`.

POST `/api/languages/`
Cria uma nova língua.

Parâmetros

- **code** (*string*) – Nome do idioma
- **name** (*string*) – Nome do idioma
- **direction** (*string*) – Direção do texto
- **plural** (*object*) – Fórmula e quantidade de plural do idioma

GET `/api/languages/(string: language) /`
Devolve informação sobre uma língua.

Parâmetros

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

Objeto JSON de resposta

- **code** (*string*) – Código do idioma
- **direction** (*string*) – Direção do texto
- **plural** (*object*) – Objeto de informações de plurais do idioma
- **aliases** (*array*) – vetor de codinomes para o idioma

Exemplo de dados JSON:

```
{
  "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*) /
Altera os parâmetros linguísticos.

Parâmetros

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

Objeto JSON da solicitação

- **name** (*string*) – Nome do idioma
- **direction** (*string*) – Direção do texto
- **plural** (*object*) – Detalhes de plural do idioma

PATCH /api/languages/ (**string:** *language*) /
Altera os parâmetros linguísticos.

Parâmetros

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

Objeto JSON da solicitação

- **name** (*string*) – Nome do idioma
- **direction** (*string*) – Direção do texto
- **plural** (*object*) – Detalhes de plural do idioma

DELETE /api/languages/ (**string:** *language*) /
Apaga o idioma.

Parâmetros

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

GET `/api/languages/ (string: language) /statistics/`
Devolve estatísticas para uma língua.

Parâmetros

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

Objeto JSON de resposta

- **total** (*int*) – quantidade total de cadeias
- **total_words** (*int*) – número total de palavras
- **last_change** (*timestamp*) – últimas alterações na língua
- **recent_changes** (*int*) – número total de alterações
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **translated_words** (*int*) – número de palavras traduzidas
- **translated_words_percent** (*int*) – percentagem de palavras traduzidas
- **translated_chars** (*int*) – número de caracteres traduzidos
- **translated_chars_percent** (*int*) – percentagem de caracteres traduzidos
- **total_chars** (*int*) – número total de caracteres
- **fuzzy** (*int*) – quantidade de cadeias aproximadas (marcados para edição)
- **fuzzy_percent** (*int*) – percentagem de cadeias aproximadas (marcados para edição)
- **failing** (*int*) – quantidade de cadeias com falha
- **failing_percent** (*int*) – percentagem de cadeias com falha

1.12.7 Projetos

GET `/api/projects/`
Retorna uma lista de todos os projetos.

Veja também:

Os atributos do objeto do projeto estão documentados em `GET /api/projects/ (string:project)/`.

POST `/api/projects/`
Novo na versão 3.9.
Cria um novo projeto.

Parâmetros

- **name** (*string*) – Nome do projeto
- **slug** (*string*) – URL amigável do projeto
- **web** (*string*) – Site da Web do Projeto

GET `/api/projects/ (string: project) /`
Devolve informação sobre um projecto.

Parâmetros

- **project** (*string*) – URL semântico do projeto

Objeto JSON de resposta

- **name** (*string*) – nome do projeto

- **slug** (*string*) – url amigável do projeto
- **web** (*string*) – site do projeto
- **components_list_url** (*string*) – URL para lista de componentes; veja `GET /api/projects/(string:project)/components/`
- **repository_url** (*string*) – URL para estado do repositório; veja `GET /api/projects/(string:project)/repository/`
- **changes_list_url** (*string*) – URL para lista de alterações; veja `GET /api/projects/(string:project)/changes/`
- **translation_review** (*boolean*) – *Activar revisões*
- **source_review** (*boolean*) – *Ativar revisões de fontes*
- **set_language_team** (*boolean*) – *Definir cabeçalho «Language-Team»*
- **enable_hooks** (*boolean*) – *Ativar hooks*
- **instructions** (*string*) – *Instruções para tradução*
- **language_aliases** (*string*) – *Aliases do idioma*

Exemplo de dados JSON:

```
{
  "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) /`
Novo na versão 4.3.

Edita um projeto por uma solicitação **PATCH**.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

PUT `/api/projects/(string: project) /`
Novo na versão 4.3.

Edita um projeto por uma solicitação `:http:method:PUT`.

Parâmetros

- **project** (*string*) – URL semântico do projeto

DELETE `/api/projects/(string: project) /`
Novo na versão 3.9.

Apaga um projeto.

Parâmetros

- **project** (*string*) – URL semântico do projeto

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

Retorna uma lista de alterações no projeto. Este é essencialmente um `GET /api/changes/` no escopo do projeto a aceitar os mesmos parâmetros.

Parâmetros

- **project** (*string*) – URL semântico do projeto

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos componentes; veja `GET /api/changes/(int:id)/`

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

Retorna informações sobre o estado do repositório VCS. Este contém apenas um resumo geral de todos os repositórios para o projeto. Para obter um estado mais detalhado, use [http://get:/api/components/\(string:project\)/\(string:component\)/repository/](http://get:/api/components/(string:project)/(string:component)/repository/).

Parâmetros

- **project** (*string*) – URL semântico do projeto

Objeto JSON de resposta

- **needs_commit** (*boolean*) – se há alguma alteração pendente de commit
- **needs_merge** (*boolean*) – se há alguma alteração do upstream para mesclar
- **needs_push** (*boolean*) – se há alguma alteração local para fazer push

Exemplo de dados JSON:

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

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

Realiza a operação dada no repositório VCS.

Parâmetros

- **project** (*string*) – URL semântico do projeto

Objeto JSON da solicitação

- **operation** (*string*) – Operação para realizar: uma de push, pull, commit, reset, cleanup, file-sync

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

Exemplo de CURL:

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

Exemplo de solicitação JSON:

```
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"}
```

Exemplo de resposta JSON:

```

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

{"result":true}

```

GET `/api/projects/(string: project)/components/`
Retorna uma lista de componentes de tradução no projeto dado.

Parâmetros

- **project** (*string*) – URL semântico do projeto

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos componentes; veja `GET /api/components/(string:project)/(string:component)/`

POST `/api/projects/(string: project)/components/`
Novo na versão 3.9.

Alterado na versão 4.3: Os parâmetros ``zipfile`` e ``docfile`` agora são aceitos para componentes sem VCS, veja *Ficheiros locais*.

Alterado na versão 4.6: The cloned repositories are now automatically shared within a project using *URLs internas do Weblate*. Use `disable_autoshare` to turn off this.

Cria componentes de tradução no projeto dado.

Dica: Use *URLs internas do Weblate* ao criar vários componentes a partir de um único repositório VCS.

Nota: A maior parte da criação de componentes acontece em segundo plano. Verifique o atributo `task_url` do componente criado e siga o progresso por lá.

Parâmetros

- **project** (*string*) – URL semântico do projeto

Parâmetros de formulário

- **file zipfile** – Ficheiro ZIP para enviar ao Weblate para inicialização de traduções
- **file docfile** – Documento para traduzir
- **boolean disable_autoshare** – Disables automatic repository sharing via *URLs internas do Weblate*.

Objeto JSON de resposta

- **result** (*object*) – Objeto componente criado; veja `GET /api/components/(string:project)/(string:component)/`

JSON não pode ser usado ao enviar os ficheiros a usar os parâmetros `zipfile` e `docfile`. Os dados devem ser enviados como *multipart/form-data*.

Exemplo de solicitação de formulário com CURL:


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

Exemplo de solicitação JSON com CURL:

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

Exemplo de solicitação JSON:

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

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "vcs": "git"
}
```

Exemplo de resposta 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
```

(continues on next page)

(continuação da página anterior)

```

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/`
 Retorna estatísticas paginadas para todos os idiomas num projeto.

Novo na versão 3.8.

Parâmetros

- **project** (*string*) – URL semântico do projeto

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de estatísticas de tradução
- **language** (*string*) – nome de idioma
- **code** (*string*) – código de idioma
- **total** (*int*) – quantidade total de cadeias
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas
- **words_percent** (*float*) – percentagem de palavras traduzidas

GET `/api/projects/(string: project)/statistics/`
 Retorna estatísticas para um projeto.

Novo na versão 3.8.

Parâmetros

- **project** (*string*) – URL semântico do projeto

Objeto JSON de resposta

- **total** (*int*) – quantidade total de cadeias
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas
- **words_percent** (*float*) – percentagem de palavras traduzidas

1.12.8 Componentes

GET `/api/components/`

Retorna uma lista de componentes de tradução.

Veja também:

Os atributos do objeto componente estão documentados em `GET /api/components/(string:project)/(string:component)/`.

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

string: *component* / Retorna informações sobre o componente de tradução.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **project** (*object*) – o projeto de tradução; veja `GET /api/projects/(string:project)/`
- **name** (*string*) – *Nome do componente*
- **slug** (*string*) – *URL amigável do componente*
- **vcs** (*string*) – *Sistema de controlo de versões*
- **repo** (*string*) – *Repositório do código-fonte*
- **git_export** (*string*) – *URL do repositório exportado*
- **branch** (*string*) – *Ramo do repositório*
- **push_branch** (*string*) – *Ramo do push*
- **filemask** (*string*) – *Máscara de ficheiros*
- **template** (*string*) – *Ficheiro de idioma base monolingue*
- **edit_template** (*string*) – *Editar ficheiro base*
- **intermediate** (*string*) – *Ficheiro de idioma intermédio*
- **new_base** (*string*) – *Modelo para novas traduções*
- **file_format** (*string*) – *Formato de ficheiro*
- **license** (*string*) – *Licença da tradução*
- **agreement** (*string*) – *Acordo de contribuidor*
- **new_lang** (*string*) – *Adicionar nova tradução*

- **language_code_style** (*string*) – *Estilo de código de idioma*
- **source_language** (*object*) – objeto do idioma fonte; veja `GET /api/languages/(string:language)/`
- **push** (*string*) – *URL de submissão do repositório*
- **check_flags** (*string*) – *Marcadores de tradução*
- **priority** (*string*) – *Prioridade*
- **enforced_checks** (*string*) – *Verificações impostas*
- **restricted** (*string*) – *Acesso restrito*
- **repoweb** (*string*) – *Navegador do repositório*
- **report_source_bugs** (*string*) – *Endereço para reportar erros na cadeia fonte*
- **merge_style** (*string*) – *Estilo de união*
- **commit_message** (*string*) – *Mensagens de commit, add, merge e extensão*
- **add_message** (*string*) – *Mensagens de commit, add, merge e extensão*
- **delete_message** (*string*) – *Mensagens de commit, add, merge e extensão*
- **merge_message** (*string*) – *Mensagens de commit, add, merge e extensão*
- **addon_message** (*string*) – *Mensagens de commit, add, merge e extensão*
- **allow_translation_propagation** (*string*) – *Permitir propagação da tradução*
- **enable_suggestions** (*string*) – *Ativar sugestões*
- **suggestion_voting** (*string*) – *Votação de sugestão*
- **suggestion_autoaccept** (*string*) – *Aceitar sugestões automaticamente*
- **push_on_commit** (*string*) – *Enviar ao submeter*
- **commit_pending_age** (*string*) – *Idade das alterações a fazer commit*
- **auto_lock_error** (*string*) – *Bloquear com erro*
- **language_regex** (*string*) – *Filtro de idioma*
- **variant_regex** (*string*) – *Expressão regular das variantes*
- **repository_url** (*string*) – URL para estado do repositório; veja `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – URL para lista de tradução; veja `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – URL para estado de bloqueio; veja `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – URL para lista de alterações; veja `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – URL para uma tarefa em segundo plano (se houver); veja `GET /api/tasks/(str:uuid)/`

Exemplo de dados JSON:

```
{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
```

(continues on next page)

```

"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/ Edita uma componente por uma solicitação **PATCH**.

Parâmetros

- **project** (string) – URL semântico do projeto
- **component** (string) – URL semântico do componente
- **source_language** (string) – Código de idioma fonte do projeto (opcional)

Objeto JSON da solicitação

- **name** (string) – nome do componente
- **slug** (string) – URL amigável do componente
- **repo** (string) – URL do repositório VCS

Exemplo de CURL:

```

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

```

Exemplo de solicitação JSON:

```

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"
}

```

Exemplo de resposta JSON:

```

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

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

```

PUT /api/components/(string: *project*) /
string: *component* / Edita um componente por uma solicitação **PUT**.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON da solicitação

- **branch** (*string*) – Ramo do repositório VCS
- **file_format** (*string*) – formato de ficheiro de traduções
- **filemask** (*string*) – máscara dos ficheiros de tradução no repositório
- **name** (*string*) – nome do componente
- **slug** (*string*) – URL amigável do componente
- **repo** (*string*) – URL do repositório VCS
- **template** (*string*) – ficheiro base para traduções monolíngues
- **new_base** (*string*) – ficheiro base para adicionar novas traduções
- **vcs** (*string*) – sistema de controle de versão

DELETE /api/components/ (**string:** *project*) /
string: *component* / Novo na versão 3.9.

Apaga um componente.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

GET /api/components/ (**string:** *project*) /
string: *component/changes/* Retorna uma lista de alterações de componentes. Este é essencialmente um componente com escop [GET /api/changes/](#) a aceitar os mesmos parâmetros.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos componentes; veja [GET /api/changes/ \(int:id\) /](#)

GET /api/components/ (**string:** *project*) /
string: *component/screenshots/* Retorna uma lista de capturas de ecrã do componente.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **results** (*array*) – vetor de capturas de ecrã do componente; veja [GET /api/screenshots/ \(int:id\) /](#)

GET /api/components/ (**string:** *project*) /
string: *component/lock/* Retorna o estado do bloqueio do componente.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **locked** (*boolean*) – se o componente está bloqueado para atualizações

Exemplo de dados JSON:

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

POST `/api/components/(string: project) /`
string: `component/lock/` Define o estado de bloqueio do componente.

A resposta é a mesma que `GET /api/components/(string:project)/(string:component)/lock/`.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON da solicitação

- **lock** – Booleano se deve ser bloqueado ou não.

Exemplo de CURL:

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

Exemplo de solicitação JSON:

```
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}
```

Exemplo de resposta 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/` Retorna informações sobre o estado do repositório VCS.

A resposta é a mesma para `GET /api/projects/(string:project)/repository/`.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **needs_commit** (*boolean*) – se há alguma alteração pendente de commit
- **needs_merge** (*boolean*) – se há alguma alteração do upstream para mesclar

- **needs_push** (*boolean*) – se há alguma alteração local para fazer push
- **remote_commit** (*string*) – Informações do commit remoto
- **status** (*string*) – Estado do repositório VCS como relatado pelo VCS
- **merge_failure** – Texto a descrever uma falha de mesclagem ou nulo se houver nenhuma

POST `/api/components/(string: project) /`
string: `component/repository/` Executa a operação fornecida num repositório VCS.

Veja `POST /api/projects/(string:project)/repository/` para a documentação.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON da solicitação

- **operation** (*string*) – Operação para executar: um de push, pull, commit, reset, cleanup

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

Exemplo de CURL:

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

Exemplo de solicitação JSON:

```
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"}
```

Exemplo de resposta 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/` Descarrega o ficheiro base para traduções monolíngues.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

GET `/api/components/(string: project) /`
string: `component/new_template/` Descarrega o ficheiro de modelo para novas traduções.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

GET `/api/components/(string: project) /`
string: `component/translations/` Retorna uma lista de objetos de tradução no componente fornecido.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de tradução; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/`

POST `/api/components/(string: project) /`
string: `component/translations/` Cria uma nova tradução no componente fornecido.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON da solicitação

- **language_code** (*string*) – código de idioma de tradução; veja `GET /api/languages/(string:language)/`

Objeto JSON de resposta

- **result** (*object*) – novo objecto de tradução criado

Exemplo de CURL:

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

Exemplo de solicitação JSON:

```
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"}
```

Exemplo de resposta 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
```

(continues on next page)

```

Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "failing_checks": 0,
  "failing_checks_percent": 0,
  "failing_checks_words": 0,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
  "is_source": false,
  "language": {
    "code": "cs",
    "direction": "ltr",
    "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/` Retorna estatísticas paginadas para todas as traduções dentro do componente.

Novo na versão 2.7.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de estatísticas de tradução; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`

GET `/api/components/(string: project) /`
string: `component/links/` Retorna projetos vinculados a um componente.

Novo na versão 4.5.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Objeto JSON de resposta

- **projects** (*array*) – projetos associados; veja `GET /api/projects/(string:project)/`

POST /api/components/(string: project) /
string: *component/links/* Associa projeto a um componente.

Novo na versão 4.5.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente

Parâmetros de formulário

- **string project_slug** – URL amigável do projeto

DELETE /api/components/(string: project) /
string: *component/links/string: project_slug/* Remove associação de um projeto com um componente.

Novo na versão 4.5.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **project_slug** (*string*) – Slug do projeto para remover

1.12.9 Traduções

GET /api/translations/
 Devolve uma lista de traduções.

Veja também:

Atributos de objeto de tradução estão documentados em `GET /api/translations/(string:project)/(string:component)/(string:language)/`.

GET /api/translations/(string: project) /
string: *component/string: language/* Devolve informação sobre uma tradução.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON de resposta

- **component** (*object*) – objeto do componente; veja `GET /api/components/(string:project)/(string:component)/`
- **failing_checks** (*int*) – quantidade de cadeias com verificações falhadas
- **failing_checks_percent** (*float*) – Cadeias traduzidas com quaisquer verificações falhadas
- **failing_checks_words** (*int*) – quantidade de verificações falhadas
- **filename** (*string*) – Nome do ficheiro da tradução
- **fuzzy** (*int*) – quantidade de cadeias aproximadas (marcados para edição)

- **fuzzy_percent** (*float*) – percentagem de cadeias aproximadas (marcados para edição)
- **fuzzy_words** (*int*) – quantidade de palavras em cadeias aproximadas (marcados para edição)
- **have_comment** (*int*) – quantidade de cadeias com comentário
- **have_suggestion** (*int*) – quantidade de cadeias com sugestão
- **is_template** (*boolean*) – se a tradução tem uma base monolíngue
- **language** (*object*) – objeto do idioma fonte; veja `GET /api/languages/(string:language)/`
- **language_code** (*string*) – código de idioma usado no repositório; isso pode ser diferente do código de idioma no objeto de linguagem
- **last_author** (*string*) – nome do último autor
- **last_change** (*timestamp*) – data e hora da última alteração
- **revision** (*string*) – hash de revisão para o ficheiro
- **share_url** (*string*) – URL para compartilhar a levar à página de engajamento
- **total** (*int*) – quantidade total de cadeias
- **total_words** (*int*) – número total de palavras
- **translate_url** (*string*) – URL para tradução
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **translated_words** (*int*) – número de palavras traduzidas
- **repository_url** (*string*) – URL para estado do repositório; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
- **file_url** (*string*) – URL para objeto ficheiro; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/file/`
- **changes_list_url** (*string*) – URL para lista de alterações; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/changes/`
- **units_list_url** (*string*) – URL para lista de cadeias; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/units/`

Exemplo de dados JSON:

```
{
  "component": {
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "new_base": "",
    "project": {
      "name": "Hello",
      "slug": "hello",
```

(continues on next page)

(continuação da página anterior)

```

    "source_language": {
      "code": "en",
      "direction": "ltr",
      "name": "English",
      "url": "http://example.com/api/languages/en/",
      "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
  },
  "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/ Novo na versão 3.9.

Apaga uma tradução.

Parâmetros

- **project** (string) – URL semântico do projeto
- **component** (string) – URL semântico do componente
- **language** (string) – Código de idioma de tradução

GET `/api/translations/(string: project) /string: component/string: language/changes/` Retorna uma lista de alterações de tradução. Este é essencialmente um `GET /api/changes/` no escopo de tradução a aceitar os mesmos parâmetros.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos componentes; veja `GET /api/changes/(int:id)/`

GET `/api/translations/(string: project) /string: component/string: language/units/` Retorna uma lista de unidades de tradução.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução
- **q** (*string*) – Texto de consulta de pesquisa; veja *Pesquisando* (opcional)

Objeto JSON de resposta

- **results** (*array*) – texto de objetos de componente; veja `GET /api/units/(int:id)/`

POST `/api/translations/(string: project) /string: component/string: language/units/` Adiciona uma nova unidade monolíngue.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON da solicitação

- **key** (*string*) – Nome da unidade de tradução
- **value** (*string*) – O valor da unidade de tradução

Veja também:

Gerir cadeias, *adding-new-strings*

POST `/api/translations/(string: project) /string: component/string: language/autotranslate/` Aciona a tradução automática.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON da solicitação

- **mode** (*string*) – Modo de tradução automático
- **filter_type** (*string*) – Tipo de filtro de tradução automática
- **auto_source** (*string*) – Automatic translation source - mt or others

- **component** (*string*) – Ativar a contribuição para a memória de tradução compartilhada para que o projeto tenha acesso a componentes adicionais.
- **engines** (*array*) – Motores de tradução automática
- **threshold** (*string*) – Limite de pontuação

GET `/api/translations/ (string: project) /`
string: *component/string: language/file/* Descarrega o ficheiro de tradução atual como está armazenado no VCS (sem o parâmetro *format*) ou convertido para outro formato (veja [Descarregar traduções](#)).

Nota: Este ponto de extremidade da API usa uma lógica diferente para a saída do resto da API, pois opera no ficheiro inteiro em vez de nos dados. O conjunto de parâmetros aceites no *format* difere e sem tal parâmetro obtém o ficheiro de tradução armazenado no VCS.

Parâmetros de consulta

- **format** – Formato de ficheiro a usar; se não for especificado nenhuma conversão de formato acontecerá; formatos de ficheiro suportados: po, mo, xiff, xiff11, tbx, csv, xlsx, json, aresource, strings

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

POST `/api/translations/ (string: project) /`
string: *component/string: language/file/* Envia um novo ficheiro com traduções.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Parâmetros de formulário

- **string conflicts** – Como lidar com conflitos (ignore, replace-translated ou replace-approved)
- **file file** – Ficheiro enviado
- **string email** – E-mail do autor
- **string author** – Nome do autor
- **string method** – Método de envio (translate, approve, suggest, fuzzy, replace, source, add), veja [Métodos de importação](#)
- **string fuzzy** – Processamento de cadeia aproximadas (marcados para edição) (*vazio*, process, approve)

Exemplo de CURL:

```
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/` Retorna informações sobre o estado do repositório VCS.

A resposta é a mesma que para `GET /api/components/(string:project)/(string:component)/repository/`.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

POST `/api/translations/(string: project) /string: component/string: language/repository/` Realiza a operação dada no repositório VCS.

Veja `POST /api/projects/(string:project)/repository/` para a documentação.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON da solicitação

- **operation** (*string*) – Operação para executar: um de push, pull, commit, re-set, cleanup

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

GET `/api/translations/(string: project) /string: component/string: language/statistics/` Devolve estatísticas detalhadas da tradução.

Novo na versão 2.7.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON de resposta

- **code** (*string*) – código de idioma
- **failing** (*int*) – número de verificações falhadas
- **failing_percent** (*float*) – percentagem de verificações com falha
- **fuzzy** (*int*) – quantidade de cadeias aproximadas (marcados para edição)
- **fuzzy_percent** (*float*) – percentagem de cadeias aproximadas (marcados para edição)
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas
- **last_author** (*string*) – nome do último autor
- **last_change** (*timestamp*) – data da última alteração
- **name** (*string*) – nome de idioma
- **total** (*int*) – quantidade total de cadeias

- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **url** (*string*) – URL para acesso à tradução (URL de engajamento)
- **url_translate** (*string*) – URL para acesso à tradução (URL real da tradução)

1.12.10 Unidades

Uma *unidade* é uma única parte de uma tradução que forma o par de um texto fonte com uma cadeia traduzida correspondente e também contém alguns metadados relacionados. O termo é derivado do [Translate Toolkit](#) e do XLIFF.

Novo na versão 2.10.

GET /api/units/

Retorna uma lista de unidades de tradução.

Veja também:

Os atributos do objeto da unidade estão documentados em `GET /api/units/(int:id)/`.

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

Alterado na versão 4.3: O `target` e `source` agora são vetores para lidar apropriadamente com cadeias plurais.

Devolve informação sobre a unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON de resposta

- **translation** (*string*) – URL de um objeto de tradução relacionado
- **source** (*array*) – cadeia fonte
- **previous_source** (*string*) – cadeia fonte anterior usado para correspondência aproximada
- **target** (*array*) – cadeia alvo
- **id_hash** (*string*) – identificador único da unidade
- **content_hash** (*string*) – identificador único da cadeia fonte
- **location** (*string*) – local da unidade no código-fonte
- **context** (*string*) – contexto da unidade de tradução
- **note** (*string*) – nota da unidade de tradução
- **flags** (*string*) – marcadores da unidade de tradução
- **state** (*int*) – estado da unidade, 0 - não traduzida, 10 - necessita tradução, 20 - traduzida, 30 - aprovada, 100 - somente leitura
- **fuzzy** (*boolean*) – se a unidade está confusa ou marcada para revisão
- **translated** (*boolean*) – Percentagem traduzido
- **approved** (*boolean*) – Tradução aprovada
- **position** (*int*) – posição da unidade no ficheiro de tradução
- **has_suggestion** (*boolean*) – Cadeia tem sugestão
- **has_comment** (*boolean*) – Cadeia tem comentário
- **has_failing_check** (*boolean*) – Cadeia tem verificação falhada

- **num_words** (*int*) – número de palavras de origem
- **priority** (*int*) – prioridade de tradução. 100 é o padrão
- **id** (*int*) – identificador da unidade
- **explanation** (*string*) – Explicação de cadeia, disponível em unidades fonte, veja [Informações adicionais sobre cadeias fonte](#)
- **extra_flags** (*string*) – Sinalizadores de cadeias adicionais, disponíveis nas unidades de fonte, veja [Personalizando comportamento a usar marcadores](#)
- **web_url** (*string*) – URL onde a unidade pode ser editada
- **souce_unit** (*string*) – Ligação da unidade fonte; veja `GET /api/units/(int:id)/`

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

Novo na versão 4.3.

Realiza uma atualização parcial na unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON da solicitação

- **state** (*int*) – estado da unidade, 0 - não traduzida, 10 - necessita edição, 20 - traduzida, 30 - aprovada (fluxo de trabalho de exigência de revisão ativado, veja [Revisores dedicados](#))
- **target** (*array*) – cadeia alvo
- **explanation** (*string*) – Explicação de cadeia, disponível em unidades fonte, veja [Informações adicionais sobre cadeias fonte](#)
- **extra_flags** (*string*) – Sinalizadores de cadeias adicionais, disponíveis nas unidades de fonte, veja [Personalizando comportamento a usar marcadores](#)

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

Novo na versão 4.3.

Realiza a atualização completa da unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON da solicitação

- **state** (*int*) – estado da unidade, 0 - não traduzida, 10 - necessita edição, 20 - traduzida, 30 - aprovada (fluxo de trabalho de exigência de revisão ativado, veja [Revisores dedicados](#))
- **target** (*array*) – cadeia alvo
- **explanation** (*string*) – Explicação de cadeia, disponível em unidades fonte, veja [Informações adicionais sobre cadeias fonte](#)
- **extra_flags** (*string*) – Sinalizadores de cadeias adicionais, disponíveis nas unidades de fonte, veja [Personalizando comportamento a usar marcadores](#)

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

Novo na versão 4.3.

Apaga a unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

1.12.11 Alterações

Novo na versão 2.10.

GET `/api/changes/`

Alterado na versão 4.1: A filtragem de alterações foi introduzida na versão 4.1.

Retorna uma lista de alterações de tradução.

Veja também:

Os atributos do objeto de alteração estão documentados em `GET /api/changes/(int:id)/`.

Parâmetros de consulta

- **user** (*string*) – Nome de utilizador do utilizador para filtros
- **action** (*int*) – Ação para filtrar, pode ser usada várias vezes
- **timestamp_after** (*timestamp*) – Data e hora no formato ISO 8601 para listar as alterações após
- **timestamp_before** (*timestamp*) – Data e hora no formato ISO 8601 para listar as alterações antes

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

Devolve informação sobre mudanças na tradução.

Parâmetros

- **id** (*int*) – ID da alteração

Objeto JSON de resposta

- **unit** (*string*) – URL de um objeto de unidade relacionado
- **translation** (*string*) – URL de um objeto de tradução relacionado
- **component** (*string*) – URL de um objeto de componente relacionado
- **user** (*string*) – URL de um objeto de utilizador relacionado
- **author** (*string*) – URL de um objeto de autor relacionado
- **timestamp** (*timestamp*) – data e hora do evento
- **action** (*int*) – identificação numérica da ação
- **action_name** (*string*) – descrição textual da ação
- **target** (*string*) – texto ou detalhe alterado de evento
- **id** (*int*) – identificador da alteração

1.12.12 Capturas de ecrã

Novo na versão 2.14.

GET `/api/screenshots/`

Retorna uma lista de informações de cadeias de capturas de ecrã.

Veja também:

Os atributos do objeto de captura de ecrã estão documentados em `GET /api/screenshots/(int:id)/`.

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

Retorna informações sobre dados de captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja `GET /api/units/(int:id)/`

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

Descarrega a imagem de captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã

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

Substitui a imagem de captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã

Parâmetros de formulário

- **file image** – Ficheiro enviado

Exemplo de CURL:

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

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

Associa cadeias fonte com uma captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã

Parâmetros de formulário

- **string unit_id** – ID da unidade

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **translation** (*string*) – URL de um objeto de tradução relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja `GET /api/units/(int:id)/`

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

int: *unit_id* Remove a associação da cadeia fonte com captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã
- **unit_id** – ID da unidade da cadeia fonte

POST `/api/screenshots/`

Cria uma nova captura de ecrã.

Parâmetros de formulário

- **file image** – Ficheiro enviado
- **string name** – Nome da captura do ecrã
- **string project_slug** – URL amigável do projeto
- **string component_slug** – URL amigável do componente
- **string language_code** – Código do idioma

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja [GET /api/screenshots/\(int:id\)/file/](#)
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja [GET /api/units/\(int:id\)/](#)

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

Edita informações parciais sobre captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja [GET /api/screenshots/\(int:id\)/file/](#)
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja [GET /api/units/\(int:id\)/](#)

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

Edita informações completas sobre captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja [GET /api/screenshots/\(int:id\)/file/](#)
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja [GET /api/units/\(int:id\)/](#)

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

Apagar captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã

1.12.13 Extensões

Novo na versão 4.4.1.

GET `/api/addons/`

Retorna uma lista de extensões.

Veja também:

Os atributos do objeto de extensão estão documentados em `GET /api/addons/(int:id)/`.

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

Retorna informações sobre dados da extensão.

Parâmetros

- **id** (*int*) – ID da extensão

Objeto JSON de resposta

- **name** (*string*) – nome de uma extensão
- **component** (*string*) – URL de um objeto de componente relacionado
- **configuration** (*object*) – Configuração opcional da extensão

Veja também:

[Extensões](#)

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

string: `component/addons/` Cria uma nova extensão.

Parâmetros

- **project_slug** (*string*) – URL amigável do projeto
- **component_slug** (*string*) – URL amigável do componente

Objeto JSON da solicitação

- **name** (*string*) – nome de uma extensão
- **configuration** (*object*) – Configuração opcional da extensão

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

Edita informações parciais sobre extensão.

Parâmetros

- **id** (*int*) – ID da extensão

Objeto JSON de resposta

- **configuration** (*object*) – Configuração opcional da extensão

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

Edita informações completas sobre extensão.

Parâmetros

- **id** (*int*) – ID da extensão

Objeto JSON de resposta

- **configuration** (*object*) – Configuração opcional da extensão

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

Apaga extensão.

Parâmetros

- **id** (*int*) – ID da extensão

1.12.14 Listas de componentes

Novo na versão 4.0.

GET `/api/component-lists/`

Retorna uma lista de listas de componentes.

Veja também:

Os atributos do objeto da lista de componentes estão documentados em `GET /api/component-lists/(str:slug)/`.

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

Retorna informações sobre a lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON de resposta

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar num painel
- **components** (*array*) – ligação para componentes associados; veja `GET /api/components/(string:project)/(string:component)/`
- **auto_assign** (*array*) – regras de atribuição automáticas

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

Altera os parâmetros da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON da solicitação

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar num painel

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

Altera os parâmetros da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON da solicitação

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar num painel

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

Apaga a lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

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

Associa um componente a uma lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Parâmetros de formulário

- **string component_id** – ID do componente

DELETE /api/component-lists/ (**str:** *slug*) /components/
str: *component_slug* Desassocia um componente da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes
- **component_slug** (*string*) – URL amigável do componente

1.12.15 Glossário

Alterado na versão 4.5: Glossários agora são armazenados como componentes, traduções e cadeias comuns. Use a respetiva API.

1.12.16 Tarefas

Novo na versão 4.4.

GET /api/tasks/
A listagem de tarefas não está disponível no momento.

GET /api/tasks/ (**str:** *uuid*) /
Retorna informações sobre uma tarefa

Parâmetros

- **uuid** (*string*) – UUID da tarefa

Objeto JSON de resposta

- **completed** (*boolean*) – Se a tarefa foi concluída
- **progress** (*int*) – Progresso da tarefa em percentagem
- **result** (*object*) – Resultado da tarefa ou detalhes do progresso
- **log** (*string*) – Registos da tarefa

1.12.17 Metrics

GET /api/metrics/
Returns server metrics.

Objeto JSON de resposta

- **units** (*int*) – Quantidade de unidades
- **units_translated** (*int*) – Quantidade de unidades traduzidas
- **users** (*int*) – Quantidade de utilizadores
- **changes** (*int*) – Quantidade de alterações
- **projects** (*int*) – Quantidade de projetos
- **components** (*int*) – Quantidade de componentes
- **translations** (*int*) – Quantidade de traduções
- **languages** (*int*) – Quantidade de idiomas usados
- **checks** (*int*) – Quantidade de verificações de qualidade acionadas

- **configuration_errors** (*int*) – Number of configuration errors
- **suggestions** (*int*) – Quantidade de sugestões pendentes
- **celery_queues** (*object*) – Lengths of Celery queues, see *Tarefas de fundo a usar o Celery*
- **name** (*string*) – Nome do servidor configurado

1.12.18 Hooks de notificação

Os ganchos de notificação permitem que aplicações externas notifiquem o Weblate que o repositório VCS foi atualizado.

Pode usar terminais de repositório para projetos, componentes e traduções para atualizar repositórios individuais; veja `POST /api/projects/(string:project)/repository/` para a documentação.

GET /hooks/update/(string: project) /
string: *component/* Obsoleto desde a versão 2.6: Use `POST /api/components/(string:project)/(string:component)/repository/` em vez disso, que funciona corretamente com autenticação para projetos limitados por ACL.

Aciona a atualização de um componente (a buscar do VCS e a procurar por alterações de tradução).

GET /hooks/update/(string: project) /
 Obsoleto desde a versão 2.6: Use `POST /api/projects/(string:project)/repository/`, que funciona corretamente com autenticação para projetos limitados por ACL.

Aciona a atualização de todos os componentes num projeto (a descarregar do VCS e a procurar por alterações de tradução).

POST /hooks/github/

Gancho especial para lidar com notificações do GitHub e atualizar automaticamente os componentes correspondentes.

Nota: O GitHub inclui suporte direto para notificar o Weblate: ative o gancho do serviço Weblate nas configurações do repositório e defina a URL para a URL da sua instalação do Weblate.

Veja também:

Receber alterações do GitHub automaticamente Para instruções sobre como configurar integração com GitHub

<https://docs.github.com/pt/github/extending-github/about-webhooks> Informações genéricas sobre Webhooks do GitHub

ENABLE_HOOKS Para ativar ganchos para todo o Weblate

POST /hooks/gitlab/

Gancho especial para lidar com notificações do GitLab e atualizar automaticamente os componentes correspondentes.

Veja também:

Receber alterações do GitLab automaticamente Para instrução sobre configurar integração com GitLab

<https://docs.gitlab.com/ce/user/project/integrations/webhooks.html> Informações genéricas sobre Webhooks do GitLab

ENABLE_HOOKS Para ativar ganchos para todo o Weblate

POST /hooks/bitbucket/

Gancho especial para lidar com notificações do Bitbucket e atualizar automaticamente os componentes correspondentes.

Veja também:

Receber alterações do Bitbucket automaticamente Para instruções sobre como configurar integração com Bitbucket

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/> Informações genéricas sobre Webhooks do Bitbucket

ENABLE_HOOKS Para ativar ganchos para todo o Weblate

POST /hooks/pagure/

Novo na versão 3.3.

Gancho especial para lidar com notificações do Pagure e atualizar automaticamente os componentes correspondentes.

Veja também:

Receber alterações do Pagure automaticamente Para instruções sobre como configurar integração com Pagure

https://docs.pagure.org/pagure/usage/using_webhooks.html Informações genéricas sobre Webhooks do Pagure

ENABLE_HOOKS Para ativar ganchos para todo o Weblate

POST /hooks/azure/

Novo na versão 3.8.

Gancho especial para lidar com notificações do Azure Repos e atualizar automaticamente os componentes correspondentes.

Veja também:

Receber alterações dos Azure Repos automaticamente Para instruções sobre como configurar integração com Azure

<https://docs.microsoft.com/pt-br/azure/devops/service-hooks/services/webhooks?view=azure-devops> Informações genéricas sobre Web Hooks do Azure Repos

ENABLE_HOOKS Para ativar ganchos para todo o Weblate

POST /hooks/gitea/

Novo na versão 3.9.

Gancho especial para lidar com notificações de Webhook do Gitea e atualizar automaticamente os componentes correspondentes.

Veja também:

Receber alterações dos Gitea Repos automaticamente Para instruções sobre como configurar integração com Gitea

<https://docs.gitea.io/pt-br/webhooks/> Informações genéricas sobre Webhooks do Gitea

ENABLE_HOOKS Para ativar ganchos para todo o Weblate

POST /hooks/gitee/

Novo na versão 3.9.

Gancho especial para lidar com notificações de Webhook do Gitee e atualizar automaticamente os componentes correspondentes.

Veja também:

Receber alterações de Gitee Repos automaticamente Para instruções sobre como configurar integração com Gitee

<https://gitee.com/help/categories/40> Informações genéricas sobre Webhooks do Gitee

ENABLE_HOOKS Para ativar ganchos para todo o Weblate

1.12.19 Exportações

O Weblate fornece várias exportações para permitir que processe ainda mais os dados.

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

Parâmetros de consulta

- **format** (*string*) – Formato de saída: json ou csv

Obsoleto desde a versão 2.6: Use `GET /api/components/(string:project)/(string:component)/statistics/` e `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/` em vez disso; ele também permite acesso a projetos controlados por ACL.

Recupera estatísticas para determinado componente em determinado formato.

Exemplo de solicitação:

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

Exemplo de resposta:

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

[
  {
    "code": "cs",
    "failing": 0,
    "failing_percent": 0.0,
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "last_author": "Michal Čihař",
    "last_change": "2012-03-28T15:07:38+00:00",
    "name": "Czech",
    "total": 436,
    "total_words": 15271,
    "translated": 436,
    "translated_percent": 100.0,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/cs/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
  },
  {
    "code": "nl",
    "failing": 21,
    "failing_percent": 4.8,
    "fuzzy": 11,
    "fuzzy_percent": 2.5,
```

(continues on next page)

```

    "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 Feeds RSS

As alterações nas traduções são exportadas em feeds RSS.

GET /exports/rss/(string: *project*) /
string: *component*/string: *language* / Recupera o feed RSS com as alterações recentes de uma tradução.

GET /exports/rss/(string: *project*) /
string: *component* / Recupera feed RSS com alterações recentes para um componente.

GET /exports/rss/(string: *project*) /
 Recupera feed RSS com alterações recentes para um projeto.

GET /exports/rss/language/(string: *language*) /
 Recupera feed RSS com alterações recentes para um idioma.

GET /exports/rss/
 Recupera feed RSS com alterações recentes para a instância Weblate.

Veja também:

[RSS na Wikipédia](#)

1.13 Cliente Weblate

Novo na versão 2.7: Há suporte total do utilitário `wlc` desde o Weblate 2.7. Se estiver a usar uma versão mais antiga, algumas incompatibilidades com a API podem ocorrer.

1.13.1 Instalação

O cliente Weblate é enviado separadamente e inclui o módulo Python. Para usar os comandos abaixo, precisa instalar `wlc`:

```
pip3 install wlc
```

1.13.2 Uso do Docker

O Weblate Client também está disponível como uma imagem Docker.

A imagem está publicada no Docker Hub: <https://hub.docker.com/r/weblate/wlc>

Instalar:

```
docker pull weblate/wlc
```

O contentor Docker usa as configurações padrão do Weblate e se conecta à API implementada no localhost. A URL da API e `API_KEY` podem ser configuradas através dos argumentos aceitos pelo Weblate.

O comando para iniciar o contentor usa a seguinte sintaxe:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Exemplo:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

Queria passar o seu *Ficheiros de configuração* para o contentor Docker, a abordagem mais fácil é adicionar o seu diretório atual como o volume `/home/weblate`:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Primeiros Passos

A configuração do `wlc` é armazenada em `~/.config/weblate` (veja *Ficheiros de configuração* para outros locais), por favor, crie-a para corresponder ao seu ambiente:

```
[weblate]
url = https://hosted.weblate.org/api/

[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Depois pode invocar comandos no servidor predefinido:

```
wlc ls
wlc commit sandbox/hello-world
```

Veja também:

Ficheiros de configuração

1.13.4 Sinopse

```
wlc [arguments] <command> [options]
```

Os comandos indicam, na verdade, qual operação deve ser realizada.

1.13.5 Descrição

O cliente Weblate é uma biblioteca Python e utilitário de linha de comando para gerir o Weblate remotamente a usar a *API REST do Weblate*. O utilitário de linha de comando pode ser invocado como **wlc** e está embutido em *wlc*.

Argumentos

O programa aceita os seguintes argumentos que definem o formato de saída ou qual a instância do Weblate a utilizar. Estes devem ser introduzidos antes de qualquer comando.

--format {csv,json,text,html}
Especifica o formato de saída.

--url URL
Especifica a URL da API. Substitui qualquer valor encontrado no ficheiro de configuração, consulte *Ficheiros de configuração*. A URL deve terminar com /api/, por exemplo, `https://hosted.weblate.org/api/`.

--key KEY
Especifica a chave do utilizador de API a ser usada. Substitui qualquer valor encontrado no ficheiro de configuração, consulte *Ficheiros de configuração*. Pode encontrar a sua chave no seu perfil no Weblate.

--config PATH
Substitui o caminho do ficheiro de configuração, consulte *Ficheiros de configuração*.

--config-section SECTION
Substitui a secção de ficheiros de configuração em uso, consulte *Ficheiros de configuração*.

Comandos

Os comandos seguintes estão disponíveis:

version
Imprime a versão atual.

list-languages
Lista os idiomas usados no Weblate.

list-projects
Lista os projetos no Weblate.

list-components
Lista os componentes no Weblate.

list-translations
Lista as traduções no Weblate.

show
Mostra o objeto do Weblate (tradução, componente ou projeto).

ls
Lista o objeto do Weblate (tradução, componente ou projeto).

commit
Faz um commit das alterações feitas num objeto Weblate (tradução, componente ou projeto).

pull

Faz um pull das alterações remotas do repositório no objeto Weblate (tradução, componente ou projeto).

push

Faz um push das alterações do objeto Weblate ao repositório remoto (tradução, componente ou projeto).

reset

Novo na versão 0.7: Suportado desde o wlc 0.7.

Redefine as alterações no objeto Weblate para corresponder ao repositório remoto (tradução, componente ou projeto).

cleanup

Novo na versão 0.9: Suportado desde o wlc 0.9.

Remove todas as alterações não rastreadas num objeto Weblate para corresponder ao repositório remoto (tradução, componente ou projeto).

repo

Exibe o estado do repositório para um determinado objeto do Weblate (tradução, componente ou projeto).

statistics

Exibe estatísticas detalhadas para um determinado objeto Weblate (tradução, componente ou projeto).

lock-status

Novo na versão 0.5: Suportado desde o wlc 0.5.

Exibe o estado do bloqueio.

lock

Novo na versão 0.5: Suportado desde o wlc 0.5.

Bloqueia o componente de tradução posterior no Weblate.

unlock

Novo na versão 0.5: Suportado desde o wlc 0.5.

Desbloqueia a tradução do componente Weblate.

changes

Novo na versão 0.7: Suportado desde o wlc 0.7 e o Weblate 2.10.

Exibe alterações para um determinado objeto.

download

Novo na versão 0.7: Suportado desde o wlc 0.7.

Descarrega um ficheiro de tradução.

--convert

Converte o formato do ficheiro, se nenhuma conversão não especificada for feita no servidor e o ficheiro for descarregado como está no repositório.

--output

Especifica o ficheiro para gravar a saída e se não for especificado é impresso na stdout (saída predefinida).

upload

Novo na versão 0.9: Suportado desde o wlc 0.9.

Descarrega um ficheiro de tradução.

--overwrite

Substitua as traduções existentes ao enviar.

--input

Ficheiro a partir do qual o conteúdo é lido, se não for especificado é lido de stdin (entrada predefinida).

Dica: Pode obter informações mais detalhadas sobre como invocar comandos individuais a passar `--help`, por exemplo: `wlc ls --help`.

1.13.6 Ficheiros de configuração

.weblate, .weblate.ini, weblate.ini Alterado na versão 1.6: Os ficheiros com extensão *.ini* também são aceites.

Por ficheiro de configuração de projeto

C:\Users\NAME\AppData\weblate.ini Novo na versão 1.6.

Ficheiro de configuração do utilizador no Windows.

~/.config/weblate Ficheiro de configuração do utilizador

/etc/xdg/weblate Ficheiro de configuração para todo o sistema

O programa segue a especificação XDG, para que possa ajustar a colocação de ficheiros de configuração por variáveis de ambiente `XDG_CONFIG_HOME` ou `XDG_CONFIG_DIRS`. No diretório do Windows `APPDATA` é o local preferido para o ficheiro de configuração.

As configurações seguintes podem ser configuradas na secção `[weblate]` (pode personalizar-lo por *--config-section*):

key

Chave de API para acessar o Weblate.

url

URL de API do servidor, a predefinição é `http://127.0.0.1:8000/api/`.

translation

Caminho à tradução predefinida - componente ou projeto.

O ficheiro de configuração é um ficheiro INI, por exemplo:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Além disso, as chaves de API podem ser armazenadas na secção `[keys]`:

```
[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Isso permite que armazene chaves nas suas configurações pessoais, enquanto usa a configuração do `.weblate` no repositório VCS para que o `wlc` saiba com qual servidor ele deve comunicar.

1.13.7 Exemplos

Imprimir a versão atual do programa:

```
$ wlc version
version: 0.1
```

Listar todos os projetos:

```
$ 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/
```

Também pode designar em qual projeto o wlc deve trabalhar:

```
$ 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/
```

Com esta configuração é fácil fazer um commit de alterações pendentes no projeto atual:

```
$ wlc commit
```

1.14 API Python do Weblate

1.14.1 Instalação

A API Python é distribuída separadamente, precisa instalar o *Cliente Weblate*: (wlc) para tê-la.

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception `wlc.WeblateException`

Classe base para todas as exceções.

Weblate

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

Parâmetros

- **key** (*str*) – Chave do utilizador
- **url** (*str*) – URL da API do servidor; se não especificado, o padrão é usado
- **config** (*wlc.config.WeblateConfig*) – Objeto de configuração, substitui quaisquer outros parâmetros.

Acessa a classe para a API, define a chave da API e, opcionalmente, a URL da API.

get (*path*)

Parâmetros **path** (*str*) – Caminho da solicitação

Tipo de retorno *object*

Executa uma única chamada de API GET.

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

Parâmetros **path** (*str*) – Caminho da solicitação

Tipo de retorno *object*

Executa uma única chamada de API GET.

1.14.3 wlc.config

WeblateConfig

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

Parâmetros **section** (*str*) – Seção de configuração para uso

Analizador de ficheiro de configuração a seguir a especificação XDG.

load (*path=None*)

Parâmetros **path** (*str*) – Caminho a partir do qual carregar a configuração.

Carrega a configuração de um ficheiro. Se nenhuma for especificada, carrega do ficheiro de configuração *wlc* (~/.config/wlc) posto no seu caminho de configuração XDG (/etc/xdg/wlc).

1.14.4 wlc.main

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

Parâmetros

- **settings** (*list*) – Configurações para substituir como lista de tuplas
- **stdout** (*object*) – objeto de ficheiro stdout para saída de impressão, usa *sys.stdout* como padrão
- **args** (*list*) – Argumentos de linha de comando para processar, usa *sys.args* como padrão

Ponto de entrada principal para interface de linha de comando.

`@wlc.main.register_command(command)`

Decorador para registrar a classe *Command* no analisador principal usado por *main()*.

Command

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

Classe principal para invocação de comandos.

2.1 Instruções de configuração

2.1.1 Instalar o Weblate

Instalando a usar Docker

Com a implantação do Weblate dockerizada, pode pôr a sua instância Weblate pessoal em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão.

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Os exemplos a seguir presumem que tem um ambiente Docker funcional, com `docker-compose` instalado. Verifique a documentação do Docker para obter instruções.

1. Clone o repositório `weblate-docker`:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Crie um ficheiro `docker-compose.override.yml` com as suas configurações. Veja *Variáveis de ambiente do Docker* para uma lista completa das variáveis de ambiente.

```
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: Se `WEBLATE_ADMIN_PASSWORD` não estiver definida, o utilizador admin é criado com uma palavra-passe aleatória mostrada na primeira inicialização.

O exemplo fornecido faz o Weblate escutar na porta 80. Edite o mapeamento da porta no ficheiro `docker-compose.override.yml` para alterar isso.

3. Inicie os contentores do Weblate:

```
docker-compose up
```

Aproveite a implantação do Weblate, ele está acessível na porta 80 do contentor `weblate`.

Alterado na versão 2.15-2: A configuração foi alterada recentemente, antes havia um contentor de servidor web separado, desde 2.15-2 o servidor web está embutido no contentor do Weblate.

Alterado na versão 3.7.1-6: Em julho de 2019 (a começar com a tag 3.7.1-6), os contentores não estão a ser executados como um utilizador root. Isso mudou a porta exposta de 80 para 8080.

Veja também:

Invocando comandos de gestão

Contentor Docker com suporte a HTTPS

Por favor, veja [Instalação](#) para instruções genéricas de implantação, esta secção apenas menciona diferenças em comparação a ela.

Usando os seus próprios certificados SSL

Novo na versão 3.8-3.

No caso de ter o seu próprio certificado SSL que deseja usar, basta pôr os ficheiros no volume de dados Weblate (veja [Volumes de contentor Docker](#)):

- `ssl/fullchain.pem` a conter o certificado, incluindo quaisquer certificados CA necessários
- `ssl/privkey.pem` a conter a chave privada

Ambos os ficheiros devem pertencer ao mesmo utilizador que inicia o contentor do docker e ter a máscara de ficheiro definida como 600 (legível e gravável apenas pelo utilizador dono).

Além disso, o contentor Weblate agora aceitará conexões SSL na porta 4443. Ainda quererá incluir o encaminhamento de porta para HTTPS na substituição de composição do docker:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

Se já hospeda outros sites no mesmo servidor, é provável que as portas 80 e 443 sejam usadas por um proxy reverso, como NGINX. Para passar a conexão HTTPS do NGINX para o contentor do docker, pode usar a seguinte configuração:

```
server {
    listen 443;
    listen [::]:443;

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

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

Substitua `<SITE_URL>`, `<SITE>` e `<EXPOSED_DOCKER_PORT>` por valores reais do seu ambiente.

Certificados SSL automáticos a usar Let's Encrypt

Caso queira usar certificados SSL [Let's Encrypt](#) gerados automaticamente na instalação pública, precisa adicionar um proxy HTTPS reverso num contentor Docker adicional, [https-portal](#) será usado para isso. Isso é usado no ficheiro `docker-compose-https.yml`. Em seguida, crie um ficheiro `docker-compose-https.override.yml` com as suas configurações:

```
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'
```

Sempre que invocar **docker-compose**, precisa passar os dois ficheiros a ele e então fazer:

```
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
```

Atualizando o contentor Docker

Normalmente, é uma boa ideia atualizar apenas o contentor Weblate e manter o contentor PostgreSQL na versão que possui, já que atualizar o PostgreSQL é muito doloroso e na maioria dos casos não traz muitos benefícios.

Pode fazer isso a manter o `docker-compose` existente e apenas obter as imagens mais recentes e reiniciar:

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

O banco de dados do Weblate deve ser migrado automaticamente na primeira inicialização e não deve haver necessidade de ações manuais adicionais.

Nota: Atualizações na 3.0 não são suportadas pelo Weblate. Se estiver na série 2.x e quiser atualizar para 3.x, primeiro atualize para a imagem 3.0.1-x mais recente (no momento em que escrevo esta é a imagem 3.0.1-7), que faça a migração e, em seguida, continue a atualizar para as versões mais recentes.

Também querará atualizar o repositório `docker-compose`, embora não seja necessário na maioria dos casos. Cuidado com as mudanças de versão do PostgreSQL neste caso, pois não é simples atualizar o banco de dados, consulte este [issue do GitHub](#) para mais informações.

Autenticação como administrador

Após a configuração do contentor, pode entrar como utilizador *admin* com a palavra-passe fornecida em `WEBLATE_ADMIN_PASSWORD`, ou uma palavra-passe aleatória gerada na primeira inicialização se não tiver sido definida.

Para redefinir a palavra-passe do *admin*, reinicie o contentor com `WEBLATE_ADMIN_PASSWORD` definido com a nova palavra-passe.

Veja também:

`WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

Quantidade de processos e consumo de memória

A quantidade de processos de trabalho para uWSGI e Celery é determinado automaticamente com base na quantidade de CPUs. Isso funciona bem para a maioria das máquinas virtuais em nuvem, pois normalmente têm poucas CPUs e boa quantidade de memória.

Caso tenha muitos núcleos de CPU e tenha problemas de memória insuficiente, tente reduzir a quantidade de workers:

```
environment:
  WEBLATE_WORKERS: 2
```

Também pode ajustar as categorias de workers individuais:

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

Veja também:

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

Dimensionando horizontalmente

Novo na versão 4.6.

Aviso: Este recurso é uma prévia da tecnologia.

Pode executar vários contentores Weblate para dimensionar o serviço horizontalmente. O volume `/app/data` deve ser compartilhado por todos os contentores, é recomendado usar um sistema de ficheiros de cluster como o GlusterFS para isso. O volume `/app/cache` deve ser separado para cada contentor.

Cada contentor Weblate tem um papel definido a usar a variável de ambiente `WEBLATE_SERVICE`. Siga atentamente a documentação, pois alguns dos serviços devem ser executados apenas uma vez no cluster e a ordem dos serviços também é importante.

Pode encontrar configuração de exemplo no repositório `docker-compose` como `docker-compose-split.yml`.

Variáveis de ambiente do Docker

Muitas das *Configurações* do Weblate podem ser definidas no contentor Docker a usar variáveis de ambiente:

Configurações genéricas

WEBLATE_DEBUG

Configura o modo de depuração do Django a usar *DEBUG*.

Exemplo:

```
environment:
  WEBLATE_DEBUG: 1
```

Veja também:

Desativar o modo de depuração

WEBLATE_LOGLEVEL

Configura o detalhamento do log.

WEBLATE_SITE_TITLE

Altera o título do site mostrado no cabeçalho de todas as páginas.

WEBLATE_SITE_DOMAIN

Configura o domínio do site. Este parâmetro é obrigatório.

Veja também:

Definir domínio correto do site, SITE_DOMAIN

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Configura o nome e o e-mail do administrador do site. É usado para *ADMINS* e para criar o utilizador *admin* (veja *WEBLATE_ADMIN_PASSWORD* para mais informações).

Exemplo:

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

Veja também:

Autenticação como administrador, Configurar administradores corretamente, ADMINS

WEBLATE_ADMIN_PASSWORD

Define a palavra-passe para o utilizador *admin*.

- Se não for definido e o utilizador *admin* não existir, ele será criado com uma palavra-passe aleatória mostrada na primeira inicialização do contentor.
- Se não for definido e o utilizador *admin* existir, nenhuma ação será executada.
- Se definido, o utilizador *admin* é ajustado em cada inicialização do contentor para corresponder a *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_NAME* e *WEBLATE_ADMIN_EMAIL*.

Aviso: Pode ser um risco de segurança armazenar a palavra-passe no ficheiro de configuração. Considere usar essa variável apenas para configuração inicial (ou deixe o Weblate gerar uma palavra-passe aleatória na inicialização) ou para recuperação de palavra-passe.

Veja também:

Autenticação como administrador, `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_PASSWORD_FILE`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

WEBLATE_ADMIN_PASSWORD_FILE

Define o caminho para um ficheiro que contém a palavra-passe para o utilizador *admin*.

Veja também:

`WEBLATE_ADMIN_PASSWORD`

WEBLATE_SERVER_EMAIL

WEBLATE_DEFAULT_FROM_EMAIL

Configura o endereço para e-mails de saída.

Veja também:

Configurar envio de e-mail

WEBLATE_CONTACT_FORM

Configura o comportamento do formulário de contato, veja *`CONTACT_FORM`*.

WEBLATE_ALLOWED_HOSTS

Configura os nomes de host HTTP permitidos a usar *`ALLOWED_HOSTS`*.

O padrão é ```*``` que permite todos os nomes de host.

Exemplo:

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

Veja também:

`ALLOWED_HOSTS`, Configuração de hosts permitidos, Definir domínio correto do site

WEBLATE_REGISTRATION_OPEN

Configura se os registos são abertos a alternar *`REGISTRATION_OPEN`*.

Exemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Configura quais métodos de autenticação podem ser usados para criar uma nova conta via *`REGISTRATION_ALLOW_BACKENDS`*.

Exemplo:

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

WEBLATE_TIME_ZONE

Configura o fuso horário usado no Weblate, veja *`TIME_ZONE`*.

Nota: Para alterar o fuso horário do próprio contentor do Docker, use a variável de ambiente ```TZ```.

Exemplo:

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

WEBLATE_ENABLE_HTTPS

Faz com que o Weblate presumira que é operado por trás de um proxy HTTPS reverso, faz com que o Weblate use HTTPS em e-mail e ligações de API ou defina sinalizadores seguros em cookies.

Dica: Por favor, consulte a documentação de [ENABLE_HTTPS](#) para possíveis advertências.

Nota: Isso não faz com que o contentor Weblate aceite conexões HTTPS, precisa configurar isso também, consulte [Contentor Docker com suporte a HTTPS](#) para exemplos.

Exemplo:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Veja também:

[ENABLE_HTTPS](#) Definir domínio correto do site, [WEBLATE_SECURE_PROXY_SSL_HEADER](#)

WEBLATE_IP_PROXY_HEADER

Permite que o Weblate obtenha o endereço IP de qualquer cabeçalho HTTP fornecido. Use isso ao usar um proxy reverso na frente do contentor Weblate.

Ativa [IP_BEHIND_REVERSE_PROXY](#) e define [IP_PROXY_HEADER](#).

Nota: O formato deve estar de acordo com as expectativas do Django. O Django [transforma](#) nomes de cabeçalho HTTP brutos da seguinte forma:

- converte todos os caracteres em maiúsculas
- substitui todos os hifenes por sublinhados
- prefixa o prefixo HTTP_

Portanto, X-Forwarded-For seria mapeado para HTTP_X_FORWARDED_FOR.

Exemplo:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Uma tupla que representa uma combinação de cabeçalho/valor HTTP que significa que uma solicitação é segura. Isso é necessário quando o Weblate está a ser executado por trás de um proxy reverso a fazer a terminação SSL que não passa cabeçalhos HTTPS padrão.

Exemplo:

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

Veja também:

[SECURE_PROXY_SSL_HEADER](#)

WEBLATE_REQUIRE_LOGIN

Ativa [REQUIRE_LOGIN](#) para impor autenticação em todo o Weblate.

Exemplo:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Adiciona exceções de URL para autenticação necessária para toda a instalação do Weblate a usar *LOGIN_REQUIRED_URLS_EXCEPTIONS*.

Pode substituir configurações inteiras ou modificar o valor padrão a usar as variáveis ADD e REMOVE.

WEBLATE_GOOGLE_ANALYTICS_ID

Configura o ID para o Google Analytics a alterar *GOOGLE_ANALYTICS_ID*.

WEBLATE_GITHUB_USERNAME

Configura o nome de utilizador do GitHub para pull requests do GitHub a alterar *GITHUB_USERNAME*.

Veja também:

GitHub

WEBLATE_GITHUB_TOKEN

Novo na versão 4.3.

Configura o token de acesso pessoal do GitHub para pull requests do GitHub via API a alterar *GITHUB_TOKEN*.

Veja também:

GitHub

WEBLATE_GITLAB_USERNAME

Configura o nome de utilizador do GitLab para merge requests do GitLab a alterar *GITLAB_USERNAME*

Veja também:

GitLab

WEBLATE_GITLAB_TOKEN

Configura o token de acesso pessoal do GitLab para merge requests do GitLab via API a alterar *GITLAB_TOKEN*

Veja também:

GitLab

WEBLATE_PAGURE_USERNAME

Configura o nome de utilizador do Pagure para merge requests do Pagure a alterar *PAGURE_USERNAME*

Veja também:

Pagure

WEBLATE_PAGURE_TOKEN

Configura o token de acesso pessoal do Pagure para merge requests do Pagure via API a alterar *PAGURE_TOKEN*

Veja também:

Pagure

WEBLATE_SIMPLIFY_LANGUAGES

Configura a política de simplificação de idioma, veja *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configura o padrão *Controlo de acesso* para novos projetos, veja *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configura o valor padrão para *Acesso restrito* para novos componentes, veja *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configura o valor padrão para *Permitir propagação da tradução* para novos componentes, veja *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Configura *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Configura *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

Configura *DEFAULT_SHARED_TM*.

WEBLATE_AKISMET_API_KEY

Configura a chave API do Akismet, veja *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Configura a assinatura GPG de commits, veja *WEBLATE_GPG_IDENTITY*.

Veja também:

Signing Git commits with GnuPG

WEBLATE_URL_PREFIX

Configura o prefixo da URL onde o Weblate está a ser executado, veja *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Configura verificações que não deseja que sejam mostradas, veja *SILENCED_SYSTEM_CHECKS*.

WEBLATE_CSP_SCRIPT_SRC**WEBLATE_CSP_IMG_SRC****WEBLATE_CSP_CONNECT_SRC****WEBLATE_CSP_STYLE_SRC****WEBLATE_CSP_FONT_SRC**

Permite personalizar o cabeçalho HTTP Content-Security-Policy.

Veja também:

Política de segurança de conteúdo, *CSP_SCRIPT_SRC*, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

WEBLATE_LICENSE_FILTER

Configura *LICENSE_FILTER*.

WEBLATE_LICENSE_REQUIRED

Configura *LICENSE_REQUIRED*

WEBLATE_WEBSITE_REQUIRED

Configura *WEBSITE_REQUIRED*

WEBLATE_HIDE_VERSION

Configura *HIDE_VERSION*.

WEBLATE_BASIC_LANGUAGES

Configura *BASIC_LANGUAGES*.

WEBLATE_DEFAULT_AUTO_WATCH

Configura *DEFAULT_AUTO_WATCH*.

WEBLATE_RATELIMIT_ATTEMPTS**WEBLATE_RATELIMIT_LOCKOUT****WEBLATE_RATELIMIT_WINDOW**

Novo na versão 4.6.

Configura o limitador de taxa.

Dica: Pode definir a configuração para qualquer escopo do limitador de taxa. Para fazer isso, adicione o prefixo `WEBLATE_` a qualquer uma das configurações descritas em *Limitação de taxa*.

Veja também:

Limitação de taxa, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

WEBLATE_ENABLE_AVATARS

Novo na versão 4.6.1.

Configura `ENABLE_AVATARS`.

Configurações de tradução automática

WEBLATE_MT_APERTIUM_APY

Ativa tradução automática do *Apertium* e define `MT_APERTIUM_APY`

WEBLATE_MT_AWS_REGION

WEBLATE_MT_AWS_ACCESS_KEY_ID

WEBLATE_MT_AWS_SECRET_ACCESS_KEY

Configura tradução automática da *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

Ativa tradução de máquina do *DeepL* e define `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

Ativa *Google Tradutor* e define `MT_GOOGLE_KEY`

WEBLATE_MT_MICROSOFT_COGNITIVE_KEY

Ativa *Tradutor dos Serviços Cognitivos da Microsoft* e define `MT_MICROSOFT_COGNITIVE_KEY`

WEBLATE_MT_MICROSOFT_ENDPOINT_URL

Define `MT_MICROSOFT_ENDPOINT_URL`. Note que isto deve conter apenas o nome de domínio.

WEBLATE_MT_MICROSOFT_REGION

Defina `MT_MICROSOFT_REGION`

WEBLATE_MT_MICROSOFT_BASE_URL

Defina `MT_MICROSOFT_BASE_URL`

WEBLATE_MT_MODERNMT_KEY

Ativa *ModernMT* e define `MT_MODERNMT_KEY`.

WEBLATE_MT_MYMEMORY_ENABLED

Ativa tradução de máquina do *MyMemory* e define `MT_MYMEMORY_EMAIL` para `WEBLATE_ADMIN_EMAIL`.

Exemplo:

```
environment:
  WEBLATE_MT_MYMEMORY_ENABLED: 1
```

WEBLATE_MT_GLOSBE_ENABLED

Ativa tradução de máquina do *Glosbe*.

```
environment:
  WEBLATE_MT_GLOSBE_ENABLED: 1
```

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED

Ativa tradução de máquina do *Serviço de Terminologia 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 tradução de máquina do *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
```

Configurações de autenticação

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**

Configuração de autenticação LDAP.

Exemplo para vinculação direta:

```
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.
```

(continues on next page)

(continuação da página anterior)

```
# another example that can be used with OpenLDAP: 'full_name:cn,email:mail'
WEBLATE_AUTH_LDAP_USER_ATTR_MAP: full_name:name,email:mail
```

Exemplo para pesquisa e vinculação:

```
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
```

Exemplo para vinculação e pesquisa de união:

```
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
```

Exemplo com pesquisar e vincular ao 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)
```

Veja também:*Autenticação por LDAP***GitHub****WEBLATE_SOCIAL_AUTH_GITHUB_KEY****WEBLATE_SOCIAL_AUTH_GITHUB_SECRET***Ativa Autenticação por GitHub.***Bitbucket****WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY****WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET***Ativa Autenticação por Bitbucket.*

Facebook

`WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY`

`WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET`

Ativa *OAuth 2 do 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`

Ativa *OAuth 2 do Google*.

GitLab

`WEBLATE_SOCIAL_AUTH_GITLAB_KEY`

`WEBLATE_SOCIAL_AUTH_GITLAB_SECRET`

`WEBLATE_SOCIAL_AUTH_GITLAB_API_URL`

Ativa *OAuth 2 do GitLab*.

Active Directory do Azure

`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY`

`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET`

Ativa autenticação por Active Directory do Azure, veja *Active Directory do Microsoft Azure*.

Active Directory do Azure com suporte a Tenant

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY`

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET`

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID`

Ativa autenticação por Active Directory do Azure com suporte a Tenant, veja *Active Directory do 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`

Ativa autenticação com Keycloak, veja a [documentação](#).

Fornecedores Linux

Pode ativar a autenticação a usar serviços de autenticação de fornecedores Linux, a definir as seguintes variáveis para qualquer valor.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Ativa a autenticação Slack, veja [Slack](#).

SAML

Chaves SAML autoassinadas são geradas automaticamente na primeira inicialização do contentor. Caso queira usar chaves próprias, ponha o certificado e a chave privada em `/app/data/ssl/saml.crt` e `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

Configurações do provedor de identidade SAML, consulte [Autenticação por SAML](#).

Outras configurações de autenticação

WEBLATE_NO_EMAIL_AUTH

Desativa autenticação por e-mail quando definido com algum valor.

Configuração de banco de dados PostgreSQL

O banco de dados é criado por `docker-compose.yml`, então essas configurações afetam os contentores Weblate e PostgreSQL.

Veja também:

[Configuração de banco de dados para o Weblate](#)

POSTGRES_PASSWORD

Palavra-passe do PostgreSQL.

POSTGRES_PASSWORD_FILE

Caminho para o ficheiro que contém a palavra-passe do PostgreSQL. Use como uma alternativa para `POSTGRES_PASSWORD`.

POSTGRES_USER

Nome de utilizador do PostgreSQL.

POSTGRES_DATABASE

Nome do banco de dados PostgreSQL.

POSTGRES_HOST

Nome de host ou endereço IP do servidor PostgreSQL. O padrão é `database`.

POSTGRES_PORT

Porta do servidor PostgreSQL. O padrão é nenhum (usa o valor padrão).

POSTGRES_SSL_MODE

Configura como o PostgreSQL lida com SSL em conexão com o servidor, para as opções possíveis, consulte [SSL Mode Descriptions](#)

POSTGRES_ALTER_ROLE

Configura o nome da função para alterar durante as migrações, consulte [Configurar Weblate para usar PostgreSQL](#).

Configurações de backup de base de dados

Veja também:

[Dados despejados para backups](#)

WEBLATE_DATABASE_BACKUP

Configura o despejo diário do banco de dados a usar [DATABASE_BACKUP](#). O padrão é `plain`.

Configuração do servidor de cache

O uso do Redis é altamente recomendado pelo Weblate e deve fornecer uma instância do Redis ao executar o Weblate no Docker.

Veja também:

[Ativar o cache](#)

REDIS_HOST

O nome de host ou endereço IP do servidor Redis. O padrão é `cache`.

REDIS_PORT

A porta do servidor Redis. O padrão é `6379`.

REDIS_DB

O número do banco de dados Redis, o padrão é `1`.

REDIS_PASSWORD

A palavra-passe do servidor Redis, não usada por padrão.

REDIS_TLS

Ativa o uso de SSL para conexão Redis.

REDIS_VERIFY_SSL

Pode ser usado para desativar a verificação de certificado SSL para conexão Redis.

Configuração do servidor de e-mail

Para fazer com que o e-mail de saída funcione, precisa fornecer um servidor de e-mail.

Exemplo de configuração TLS:

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

Exemplo de configuração 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
```

Veja também:

Configuração de e-mail de saída

WEBLATE_EMAIL_HOST

Nome de host ou endereço IP do servidor de correio.

Veja também:

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

WEBLATE_EMAIL_PORT

Porta do servidor de correio, o padrão é 25.

Veja também:

`EMAIL_PORT`

WEBLATE_EMAIL_HOST_USER

Utilizador da autenticação por e-mail.

Veja também:

`EMAIL_HOST_USER`

WEBLATE_EMAIL_HOST_PASSWORD

Palavra-passe da autenticação por e-mail.

Veja também:

`EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Caminho para o ficheiro que contém a palavra-passe da autenticação por e-mail.

Veja também:

`WEBLATE_EMAIL_HOST_PASSWORD`

WEBLATE_EMAIL_USE_SSL

Se deve usar uma conexão TLS (segura) implícita ao falar com o servidor SMTP. Na maioria das documentações de e-mail, esse tipo de conexão TLS é conhecido como SSL. Geralmente é usado na porta 465. Se estiver a ter problemas, consulte a configuração TLS explícita `WEBLATE_EMAIL_USE_TLS`.

Veja também:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_TLS`, `EMAIL_USE_SSL`

WEBLATE_EMAIL_USE_TLS

Se deve usar uma conexão TLS (segura) ao falar com o servidor SMTP. Isso é usado para conexões TLS explícitas, geralmente na porta 587 ou 25. Se estiver a ter conexões travadas, consulte a configuração TLS implícita `WEBLATE_EMAIL_USE_SSL`.

Veja também:

`WEBLATE_EMAIL_PORT`, `WEBLATE_EMAIL_USE_SSL`, `EMAIL_USE_TLS`

WEBLATE_EMAIL_BACKEND

Configura o back-end do Django para usar no envio de e-mails.

Veja também:*Configurar envio de e-mail*, `EMAIL_BACKEND`**Integração do site****WEBLATE_GET_HELP_URL**Configura `GET_HELP_URL`.**WEBLATE_STATUS_URL**Configura `STATUS_URL`.**WEBLATE_LEGAL_URL**Configura `LEGAL_URL`.**Relatório de erro**

É recomendado coletar erros da instalação sistematicamente, veja *Coletando relatórios de erros*.

Para ativar o suporte para Rollbar, defina o seguinte:

ROLLBAR_KEY

O seu token de acesso ao servidor de postagem Rollbar.

ROLLBAR_ENVIRONMENTO seu ambiente Rollbar, o padrão é `production`.

Para ativar o suporte para Sentry, defina o seguinte:

SENTRY_DSN

O seu DSN no Sentry.

SENTRY_ENVIRONMENT

O seu ambiente no Sentry (opcional).

CDN de localização**WEBLATE_LOCALIZE_CDN_URL****WEBLATE_LOCALIZE_CDN_PATH**

Novo na versão 4.2.1.

Configuração para *CDN de localização JavaScript*.

O `WEBLATE_LOCALIZE_CDN_PATH` é o caminho dentro do contentor. Ele deve ser armazenado no volume persistente e não no armazenamento temporário.

Uma das possibilidades é armazenar isso dentro do diretório de dados do Weblate:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Nota: É responsável por configurar o serviço dos ficheiros gerados pelo Weblate, ele só armazena os ficheiros no local configurado.

Veja também:`weblate-cdn`, `LOCALIZE_CDN_URL`, `LOCALIZE_CDN_PATH`

Alterando apps, verificações, complementos ou correções automáticas ativados

Novo na versão 3.8-5.

A configuração embutida de verificações, complementos ou correções automática ativados pode ser ajustada pelas seguintes variáveis:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Exemplo:

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

Veja também:

CHECK_LIST, *AUTOFIX_LIST*, *WEBLATE_ADDONS*, *INSTALLED_APPS*

Configurações do contentor

WEBLATE_WORKERS

Novo na versão 4.6.1.

Quantidade base de processos de trabalho em execução no contentor. Quando não definido, é determinado automaticamente na inicialização do contentor com base na quantidade de núcleos de CPU disponíveis.

É usado para determinar *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS* e *UWSGI_WORKERS*. Pode usar essas configurações para fazer o ajuste fino.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

Essas variáveis permitem que ajuste as opções do worker do Celery. Pode ser útil ajustar a simultaneidade (`--concurrency 16`) ou usar diferentes implementações de pool (`--pool=gevent`).

Por padrão, a quantidade de workers simultâneos é baseado em *WEBLATE_WORKERS*.

Exemplo:

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

Veja também:

Opções de worker do Celery, *Tarefas de fundo a usar o Celery*

UWSGI_WORKERS

Configura quantos workers uWSGI devem ser executados.

O padrão é `WEBLATE_WORKERS`.

Exemplo:

```
environment:
  UWSGI_WORKERS: 32
```

WEBLATE_SERVICE

Define quais serviços devem ser executados dentro do contentor. Use isto para *Dimensionando horizontalmente*.

Os seguintes serviços são definidos:

celery-beat Agendador de tarefas do Celery, apenas uma instância deve estar em execução. Este contentor também é responsável pelas migrações da estrutura do banco de dados e deve ser iniciado antes dos demais.

celery-backup Worker do Celery para backups, apenas uma instância deve estar em execução.

celery-celery Worker genérico do Celery.

celery-memory Worker do Celery para memória de tradução.

celery-notify Worker do Celery para notificações.

celery-translate Worker do Celery para tradução automática.

web Servidor web.

Volumes de contentor Docker

Há dois volumes (dados e cache) exportados pelo contentor Weblate. Os outros contentores de serviço (PostgreSQL ou Redis) também têm os volumes de dados deles, mas eles não são cobertos por este documento.

O volume de dados é usado para armazenar dados persistentes do Weblate, como repositórios clonados ou para personalizar a instalação do Weblate.

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`.

O volume do cache é montado como `/app/cache` e é usado para armazenar ficheiros estáticos. O conteúdo deles é recriado na inicialização do contentor e o volume pode ser montado a usar um sistema de ficheiros efêmero como `tmpfs`.

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

Veja também:

[Documentação sobre volumes do Docker](#)

Personalização adicional da configuração

Pode personalizar ainda mais a instalação do Weblate no volume de dados, veja *Volumes de contentor Docker*.

Ficheiros de configuração personalizados

Também pode sobrescrever a configuração em `/app/data/settings-override.py` (veja *Volumes de contentor Docker*). Isso é executado no final das configurações embutidas, depois que todas as configurações de ambiente são carregadas e pode ajustá-las ou substituí-las.

Substituindo o logotipo e outros ficheiros estáticos

Novo na versão 3.8-5.

Os ficheiros estáticos que vêm com Weblate podem ser sobrescritos a colocar em `/app/data/python/customize/static` (veja *Volumes de contentor Docker*). Por exemplo, criar `/app/data/python/customize/static/favicon.ico` substituirá o `favicon`.

Dica: Os ficheiros são copiados para o local correspondente na inicialização do contentor, portanto, é necessário reiniciar o Weblate após alterar o conteúdo do volume.

Como alternativa, também pode incluir o próprio módulo (veja `../customize`) e adicioná-lo como um volume separado ao contentor do Docker, por exemplo:

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

Adicionando os seus próprios módulos Python

Novo na versão 3.8-5.

Pode pôr os próprios módulos Python em `/app/data/python/` (veja *Volumes de contentor Docker*) e eles podem ser carregados pelo Weblate, provavelmente a usar `docker-custom -config`.

Veja também:

Personalizar o Weblate

Selecione a sua máquina – provedores locais ou na nuvem

Com Docker Machine, pode criar a sua implantação Weblate na sua máquina local ou em qualquer grande quantidade de implantações baseadas em nuvem, por exemplo, Amazon AWS, Greenhost e muitos outros provedores.

Instalando no Debian e Ubuntu

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
apt install \
  libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-dev \
  libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev libacl1-dev libssl-dev \
  build-essential python3-gdbm python3-dev python3-pip python3-virtualenv
↪virtualenv git
```

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
apt install tesseract-ocr libtesseract-dev liblptonica-dev
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação local:

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

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

# Caching backend: Redis
apt install redis-server

# Database server: PostgreSQL
apt install postgresql postgresql-contrib
```

(continues on next page)

(continuação da página anterior)

```
# SMTP server
apt install exim4
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique [virtualenv user_guide](#).

1. Crie o virtualenv para Weblate:

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

2. Ative o virtualenv para Weblate:

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

3. Instale o Weblate incluindo todas as dependências:

```
pip install Weblate
```

4. Instale o driver do banco de dados:

```
pip install psycopg2-binary
```

5. Instale as dependências opcionais desejadas a depender dos recursos que pretende usar (alguns podem exigir bibliotecas de sistema adicionais, confira [Dependências opcionais](#)):

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

Configurando o Weblate

Nota: Os passos seguintes assumem que o virtualenv usado pelo Weblate está ativo (o que pode ser feito por `. ~/weblate-env/bin/activate`). Caso isso não seja verdade, terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Pode ficar com o exemplo enviado para fins de teste, mas quererá mudanças para configuração de produção, veja [Ajustar a configuração](#).
3. Crie o banco de dados e a estrutura dele para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para configuração pronta para produção):

```
weblate migrate
```

4. Crie a conta de utilizador do administrador e copie a palavra-passe de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete ficheiros estáticos para o servidor web (veja [Executar o servidor](#) e [Servir ficheiros estáticos](#)):

```
weblate collectstatic
```

6. Compactar ficheiros JavaScript e CSS (opcional, veja [Comprimir os ativos do cliente](#)):

```
weblate compress
```

7. Inicie workers de Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. Veja [Tarefas de fundo a usar o Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executar o servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registe-se com novos utilizadores.
- Agora pode executar comandos Weblate a usar o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou a usar **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: nome do componente, endereço do repositório VCS e máscara para localizar ficheiros traduzíveis. Weblate suporta uma ampla gama de formatos, a incluir ficheiros PO gettext, cadeias de recursos Android, propriedades de cadeias iOS, propriedades Java ou ficheiros de Qt Linguist, consulte [Formatos de ficheiros suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando no SUSE e openSUSE

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
zypper install \
    libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel libyaml-
↪devel \
    cairo-devel typelib-1_0-Pango-1_0 gobject-introspection-devel libacl-devel \
    python3-pip python3-virtualenv python3-devel git
```

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação local:

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

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

# Caching backend: Redis
zypper install redis-server

# Database server: PostgreSQL
zypper install postgresql postgresql-contrib
```

(continues on next page)

(continuação da página anterior)

```
# SMTP server
zypper install postfix
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique `virtualenv user_guide`.

1. Crie o virtualenv para Weblate:

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

2. Ative o virtualenv para Weblate:

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

3. Instale o Weblate incluindo todas as dependências:

```
pip install Weblate
```

4. Instale o driver do banco de dados:

```
pip install psycopg2-binary
```

5. Instale as dependências opcionais desejadas a depender dos recursos que pretende usar (alguns podem exigir bibliotecas de sistema adicionais, confira [Dependências opcionais](#)):

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

Configurando o Weblate

Nota: Os passos seguintes assumem que o virtualenv usado pelo Weblate está ativo (o que pode ser feito por `. ~/weblate-env/bin/activate`). Caso isso não seja verdade, terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Pode ficar com o exemplo enviado para fins de teste, mas quererá mudanças para configuração de produção, veja [Ajustar a configuração](#).
3. Crie o banco de dados e a estrutura dele para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para configuração pronta para produção):

```
weblate migrate
```

4. Crie a conta de utilizador do administrador e copie a palavra-passe de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete ficheiros estáticos para o servidor web (veja [Executar o servidor](#) e [Servir ficheiros estáticos](#)):

```
weblate collectstatic
```

6. Compactar ficheiros JavaScript e CSS (opcional, veja [Comprimir os ativos do cliente](#)):

```
weblate compress
```

7. Inicie workers de Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. Veja [Tarefas de fundo a usar o Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executar o servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registe-se com novos utilizadores.
- Agora pode executar comandos Weblate a usar o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou a usar **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: nome do componente, endereço do repositório VCS e máscara para localizar ficheiros traduzíveis. Weblate suporta uma ampla gama de formatos, a incluir ficheiros PO gettext, cadeias de recursos Android, propriedades de cadeias iOS, propriedades Java ou ficheiros de Qt Linguist, consulte [Formatos de ficheiros suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando no RedHat, Fedora e CentOS

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
dnf install \
    libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel libyaml-
    ↪devel \
    cairo-devel pango-devel gobject-introspection-devel libacl-devel \
    python3-pip python3-virtualenv python3-devel git
```

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação local:

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

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

# Caching backend: Redis
dnf install redis

# Database server: PostgreSQL
dnf install postgresql postgresql-contrib
```

(continues on next page)

(continuação da página anterior)

```
# SMTP server
dnf install postfix
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique [virtualenv user_guide](#).

1. Crie o virtualenv para Weblate:

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

2. Ative o virtualenv para Weblate:

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

3. Instale o Weblate incluindo todas as dependências:

```
pip install Weblate
```

4. Instale o driver do banco de dados:

```
pip install psycopg2-binary
```

5. Instale as dependências opcionais desejadas a depender dos recursos que pretende usar (alguns podem exigir bibliotecas de sistema adicionais, confira [Dependências opcionais](#)):

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

Configurando o Weblate

Nota: Os passos seguintes assumem que o virtualenv usado pelo Weblate está ativo (o que pode ser feito por `. ~/weblate-env/bin/activate`). Caso isso não seja verdade, terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Pode ficar com o exemplo enviado para fins de teste, mas quererá mudanças para configuração de produção, veja [Ajustar a configuração](#).
3. Crie o banco de dados e a estrutura dele para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para configuração pronta para produção):

```
weblate migrate
```

4. Crie a conta de utilizador do administrador e copie a palavra-passe de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete ficheiros estáticos para o servidor web (veja [Executar o servidor](#) e [Servir ficheiros estáticos](#)):

```
weblate collectstatic
```

6. Compactar ficheiros JavaScript e CSS (opcional, veja [Comprimir os ativos do cliente](#)):

```
weblate compress
```

7. Inicie workers de Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. Veja [Tarefas de fundo a usar o Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executar o servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registe-se com novos utilizadores.
- Agora pode executar comandos Weblate a usar o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou a usar **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: nome do componente, endereço do repositório VCS e máscara para localizar ficheiros traduzíveis. Weblate suporta uma ampla gama de formatos, a incluir ficheiros PO gettext, cadeias de recursos Android, propriedades de cadeias iOS, propriedades Java ou ficheiros de Qt Linguist, consulte [Formatos de ficheiros suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando no macOS

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip3 install virtualenv
```

Certifique-se de que o pip será capaz de encontrar a versão do libffi fornecida pelo homebrew – isso será necessário durante a etapa de compilação da instalação.

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

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
brew install tesseract
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação 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
```

(continues on next page)

(continuação da página anterior)

```
# Database server: PostgreSQL
brew install postgresql
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique [virtualenv user_guide](#).

1. Crie o virtualenv para Weblate:

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

2. Ative o virtualenv para Weblate:

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

3. Instale o Weblate incluindo todas as dependências:

```
pip install Weblate
```

4. Instale o driver do banco de dados:

```
pip install psycopg2-binary
```

5. Instale as dependências opcionais desejadas a depender dos recursos que pretende usar (alguns podem exigir bibliotecas de sistema adicionais, confira [Dependências opcionais](#)):

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

Configurando o Weblate

Nota: Os passos seguintes assumem que o virtualenv usado pelo Weblate está ativo (o que pode ser feito por `. ~/weblate-env/bin/activate`). Caso isso não seja verdade, terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.7/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.7/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Pode ficar com o exemplo enviado para fins de teste, mas quererá mudanças para configuração de produção, veja [Ajustar a configuração](#).
3. Crie o banco de dados e a estrutura dele para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para configuração pronta para produção):

```
weblate migrate
```

4. Crie a conta de utilizador do administrador e copie a palavra-passe de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete ficheiros estáticos para o servidor web (veja [Executar o servidor](#) e [Servir ficheiros estáticos](#)):

```
weblate collectstatic
```

6. Compactar ficheiros JavaScript e CSS (opcional, veja [Comprimir os ativos do cliente](#)):

```
weblate compress
```

7. Inicie workers de Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. Veja [Tarefas de fundo a usar o Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executar o servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registe-se com novos utilizadores.
- Agora pode executar comandos Weblate a usar o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou a usar **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: nome do componente, endereço do repositório VCS e máscara para localizar ficheiros traduzíveis. Weblate suporta uma ampla gama de formatos, a incluir ficheiros PO gettext, cadeias de recursos Android, propriedades de cadeias iOS, propriedades Java ou ficheiros de Qt Linguist, consulte [Formatos de ficheiros suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando a partir do código-fonte

1. Siga as instruções de instalação do seu sistema primeiro:

- [Instalando no Debian e Ubuntu](#)
- [Instalando no SUSE e openSUSE](#)
- [Instalando no RedHat, Fedora e CentOS](#)

2. Pegue o código-fonte mais recente do Weblate a usar Git (ou baixe um tarball e descompacte-o):

```
git clone https://github.com/WeblateOrg/weblate.git weblate-src
```

Alternativamente, pode usar ficheiros lançados. Pode descarregá-los no nosso site <<https://weblate.org/>>. Essas descarregas são assinadas criptograficamente, consulte [Verificar assinaturas de lançamento](#).

3. Instale o código atual do Weblate no virtualenv:

```
. ~/weblate-env/bin/activate
pip install -e weblate-src
```

4. Copie `weblate/settings_example.py` para `weblate/settings.py`.
5. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Pode ficar com o exemplo enviado para fins de teste, mas quererá mudanças para configuração de produção, veja [Ajustar a configuração](#).
6. Crie o banco de dados usado pelo Weblate, veja [Configuração de banco de dados para o Weblate](#).
7. Construa tabelas Django, ficheiros estáticos e dados iniciais (veja [Preencher o banco de dados](#) e [Servir ficheiros estáticos](#)):

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

Nota: Esta etapa deve ser repetida sempre que atualizar o repositório.

Instalando no OpenShift

Com o modelo OpenShift Weblate, pode pôr a sua instância pessoal do Weblate em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão e declarações de volume persistentes são usadas.

Pode encontrar o modelo em <<https://github.com/WeblateOrg/openshift/>>.

Instalação

Os exemplos a seguir presumem que tem um ambiente OpenShift v3.x funcional, com a ferramenta cliente `oc` instalada. Confira a documentação do OpenShift para obter instruções.

O `template.yml` é adequado para executar todos os componentes em OpenShift. Existe também `template-external-postgresql.yml` que não inicia um servidor PostgreSQL e permite que configure um servidor PostgreSQL externo.

Console web

Copie o conteúdo bruto de `template.yml` e importe-o ao seu projeto, então use o botão `Create` no console web do OpenShift para criar a sua aplicação. O console web solicitará os valores de todos os parâmetros usados pelo modelo.

CLI

Para enviar o modelo Weblate para a biblioteca de modelos do seu projeto atual, passe o ficheiro `template.yml` com o seguinte comando:

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

O modelo agora está disponível para seleção a usar o console web ou o CLI.

Parâmetros

Os parâmetros que pode substituir estão listados na secção de parâmetros do modelo. Pode listá-los com a CLI a usar o seguinte comando e a especificar o ficheiro a ser usado:

```
$ 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
```

Provisionamento

Também pode usar a CLI para processar modelos e usar a configuração gerada para criar objetos imediatamente.

```
$ 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
```

A instância do Weblate deve estar disponível após a migração e implantação com sucesso no parâmetro `WEBLATE_SITE_DOMAIN` especificado.

Após a configuração do contentor, pode entrar como utilizador *admin* com a palavra-passe fornecida em `WEBLATE_ADMIN_PASSWORD`, ou uma palavra-passe aleatória gerada na primeira inicialização se não tiver sido definida.

Para redefinir a palavra-passe do *admin*, reinicie o contentor com `WEBLATE_ADMIN_PASSWORD` definido para a nova palavra-passe no respectivo `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
```

Configuração

Ao processar o modelo, um respectivo ConfigMap será criado e poderá ser usado para personalizar a imagem Weblate. O ConfigMap é montado diretamente como variáveis de ambiente e dispara uma nova implantação toda vez que é alterado. Para mais opções de configuração, consulte *Variáveis de ambiente do Docker* para uma lista completa de variáveis de ambiente.

Instalando no Kubernetes

Nota: Este guia procura colaboradores experientes com o Kubernetes para abordar a configuração com mais detalhes.

Com o chart de Kubernetes Helm, pode pôr a sua instância pessoal do Weblate em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão e declarações de volume persistentes são usadas.

Pode encontrar o chart em <<https://github.com/WeblateOrg/helm/>> e pode ser exibido em <<https://artifacthub.io/packages/helm/weblate/weblate>>.

Instalação

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Configuração

For further configuration options, see *Variáveis de ambiente do Docker* for full list of environment variables.

Dependendo da sua configuração e experiência, escolha um método de instalação apropriado para si:

- *Instalando a usar Docker*, recomendado para configurações de produção.
- Instalação virtualenv, recomendada para configurações de produção:
 - *Instalando no Debian e Ubuntu*
 - *Instalando no SUSE e openSUSE*
 - *Instalando no RedHat, Fedora e CentOS*
 - *Instalando no macOS*

- *Instalando a partir do código-fonte*, recomendado para o desenvolvimento.
- *Instalando no OpenShift*
- *Instalando no Kubernetes*

2.1.2 Requisitos de software

Sistema operacional

Weblate é conhecido por funcionar no Linux, FreeBSD e macOS. Outros sistemas como o Unix provavelmente funcionarão também.

O Weblate não é suportado no Windows. Mas ainda pode funcionar e patches são aceitos alegremente.

Outros serviços

Weblate está a usar outros serviços para a operação dele. Precisarão pelo menos os seguintes serviços em execução:

- Servidor de banco de dados PostgreSQL, consulte *Configuração de banco de dados para o Weblate*.
- Servidor Redis para cache e fila de tarefas, consulte *Tarefas de fundo a usar o Celery*.
- Servidor SMTP para e-mails de saída, consulte *Configuração de e-mail de saída*.

Dependências Python

Weblate é escrito em Python e tem suporte de Python 3.6 ou mais novo. Pode instalar dependências a usar pip ou dos pacotes de distribuição deles, a lista completa está disponível em `requirements.txt`.

As dependências mais notáveis:

Django <https://www.djangoproject.com/>

Celery <https://docs.celeryproject.org/>

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

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

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

Django REST Framework <https://www.django-rest-framework.org/>

Dependências opcionais

Os módulos seguintes são necessários para alguns recursos do Weblate. Pode encontrar todos em `requirements-optional.txt`.

Mercurial (opcional para suporte de repositórios Mercurial) <https://www.mercurial-scm.org/>

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

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

akismet (opcional para a sugestão de proteção de spam) <https://github.com/ubernostrum/akismet>

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

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

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

Dependências de backend de banco de dados

O Weblate tem suporte de PostgreSQL, MySQL e MariaDB, consulte *Configuração de banco de dados para o Weblate* e a documentação dos backends para mais detalhes.

Outros requisitos do sistema

As dependências seguintes devem ser instaladas no sistema:

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

Pango, Cairo e ficheiros de cabeçalho relacionados e dados de introspecção gir <https://cairographics.org/>, <https://pango.gnome.org/>, veja *Pango e Cairo*

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

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

tesseract e os dados dele (opcional para OCR de capturas de ecrã) <https://github.com/tesseract-ocr/tesseract>

licensee (opcional para detetar a licença ao criar o componente) <https://github.com/licensee/licensee>

Dependências de tempo de compilação

Para compilar alguns das *dependências Python*, pode precisar instalar as dependências deles. Isso depende de como instala-as, por isso consulte pacotes individuais para obter documentação. Não precisará deles se usar *Wheels* pré-compilado durante a instalação a usar *pip* ou quando usar pacotes de distribuição.

Pango e Cairo

Alterado na versão 3.7.

O Weblate usa Pango e Cairo para renderizar widgets de bitmap (ver *promotion*) e verificações de renderização (ver *Gerir letras*). Para instalar as ligações Python corretamente para esses, precisa de instalar bibliotecas de sistemas primeiro - precisa tanto do Cairo quanto do Pango, que por sua vez precisam de GLib. Todos esses devem ser instalados com ficheiros de desenvolvimento e dados de introspecção GObject.

2.1.3 Verificar assinaturas de lançamento

Os lançamentos do Weblate são criptograficamente assinados pelo programador que os lançou. Atualmente é Michal Čihař. A impressão digital da chave PGP é:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

e pode obter mais informações de identificação de <https://keybase.io/nijel>.

Deve verificar se a assinatura corresponde ao ficheiro que descarregou. Desta forma, pode ter certeza de que está a usar o mesmo código que foi lançado. Também deve verificar a data da assinatura para ter certeza de que descarregou a versão mais recente.

Cada arquivo é acompanhado de ficheiros `.asc`, os quais contêm a assinatura PGP para ele. Uma vez que tenha ambos na mesma pasta, pode verificar a assinatura:

```
$ 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 pode ver, o GPG reclama que não conhece a chave pública. Neste ponto deve fazer um dos seguintes passos:

- Use *wkd* para descarregar a chave:

```
$ gpg --auto-key-locate wkd --locate-keys michal@cihar.com
pub   rsa4096 2009-06-17 [SC]
      63CB1DF1EF12CF2AC0EE5A329C27B31342B7511D
uid           [ultimate] Michal Čihař <michal@cihar.com>
uid           [ultimate] Michal Čihař <nijel@debian.org>
uid           [ultimate] [jpeg image of size 8848]
uid           [ultimate] Michal Čihař (Braiiins) <michal.cihar@braiiins.cz>
sub   rsa4096 2009-06-17 [E]
sub   rsa4096 2015-09-09 [S]
```

- Descarregue o chaveiro do [servidor do Michal](#) e importe-o com:

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

- Descarregue e importe a chave de um dos servidores principais:

```
$ 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
```

Isso vai melhorar a situação um pouco - neste momento pode verificar que a assinatura da chave dada está correta, mas ainda não pode confiar no nome usado na chave:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:      using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:      aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:      aka "[jpeg image of size 8848]" [ultimate]
gpg:      aka "Michal Čihař (Braiiins) <michal.cihar@braiiins.cz>"_
↪[ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:      There is no indication that the signature belongs to the owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

O problema aqui é que qualquer um poderia emitir a chave com este nome. Precisa garantir que a chave é realmente a propriedade da pessoa mencionada. O Manual de Privacidade do GNU aborda este tópico no capítulo [Validating other keys on your public keyring](#). O método mais confiável é de conhecer o programador pessoalmente e trocar impressões digitais importantes, no entanto também pode confiar na rede de confiança. Dessa forma, pode confiar na chave transitivamente através de assinaturas de outros, que conheceram o programador pessoalmente.

Uma vez que a chave seja confiável, o aviso não ocorrerá:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar 3 16:43:15 2019 CET
gpg:      using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:      aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:      aka "[jpeg image of size 8848]" [ultimate]
gpg:      aka "Michal Čihař (Braiiins) <michal.cihar@braiiins.cz>"_
↪[ultimate]
```

Se a assinatura for inválida (o ficheiro foi alterado), obterá um erro claro, independentemente do fato de que a chave é confiável ou não:

```
$ 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 Permissões do sistema de ficheiros

O processo Weblate precisa ser capaz de ler e escrever para o diretório onde mantém os dados – *DATA_DIR*. Todos os ficheiros dentro deste diretório devem ser de propriedade e graváveis pelo utilizador que executa todos os processos do Weblate (geralmente WSGI e Celery, veja *Executar o servidor* e *Tarefas de fundo a usar o Celery*).

A configuração predefinida põe-os na mesma árvore que as fontes do Weblate, no entanto, pode preferir movê-los para um local melhor, como `/var/lib/weblate`.

O Weblate tenta criar esses diretórios automaticamente, mas ele falhará quando não tiver permissões para fazê-lo.

Também deve tomar cuidado ao executar *Comandos de gerência*, pois eles devem ser executados sob o mesmo utilizador que o Weblate em si está a ser executado, caso contrário, permissões em alguns ficheiros podem estar erradas.

No contentor Docker, todos os ficheiros no volume `/app/data` tem de ter como dono o utilizador weblate dentro do contentor (UID 1000).

Veja também:

Servir ficheiros estáticos

2.1.5 Configuração de banco de dados para o Weblate

Recomenda-se a executar o Weblate com um servidor de banco de dados PostgreSQL.

Veja também:

Usar um poderoso mecanismo de banco de dados, Databases, Migrando de outros bancos de dados para o PostgreSQL

PostgreSQL

PostgreSQL é geralmente a melhor escolha para sites baseados em Django. É o banco de dados de referência usado para implementar a camada de banco de dados Django.

Nota: O Weblate usa a extensão trigram que deve ser instalada separadamente em alguns casos. Procure por `postgresql-contrib` ou um pacote com nome similar.

Veja também:

PostgreSQL notes

Criar um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate num banco de dados separado e separar a conta do utilizador:

```
# 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
```

Dica: Se não quiser fazer do utilizador do Weblate um superutilizador no PostgreSQL, pode omiti-lo. Nesse caso, terá que executar algumas das etapas de migração manualmente como um superutilizador do PostgreSQL no esquema Weblate usará:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
```

Configurar Weblate para usar PostgreSQL

O trecho `settings.py` para PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

A migração do banco de dados executa `ALTER ROLE` na função de banco de dados usada pelo Weblate. Na maioria dos casos, o nome da função corresponde ao nome de utilizador. Em configurações mais complexas, o nome da função é diferente do nome de utilizador e obterá um erro sobre a função não existente durante a migração do banco de dados (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Isso é conhecido por acontecer com o Azure Database para PostgreSQL, mas não está limitado a este ambiente. Defina `ALTER_ROLE` para alterar o nome da função que o Weblate deve alterar durante a migração do banco de dados.

MySQL e MariaDB

Dica: Alguns recursos do Weblate terão melhor desempenho com *PostgreSQL*. Isso inclui a memória de pesquisa e tradução, que ambos utilizam recursos de texto completo no banco de dados e a implementação do PostgreSQL é superior.

Weblate também pode ser usado com MySQL ou MariaDB, consulte [MySQL notes](#) e [MariaDB notes](#) para ressalvas ao uso do Django com elas. or causa das limitações, é recomendado usar *PostgreSQL* para novas instalações.

O Weblate requer MySQL pelo menos 5.7.8 ou MariaDB pelo menos 10.2.7.

A configuração seguinte é recomendada para Weblate:

- Use o conjunto de caracteres `utf8mb4` para permitir a representação de planos Unicode mais altos (por exemplo, emojis).
- Configure o servidor com `innodb_large_prefix` para permitir índices mais longos em campos de texto.
- Defina o nível de isolamento para `READ COMMITTED`.

- O modo SQL deve ser definido como `STRICT_TRANS_TABLES`.

MySQL 8.x, MariaDB 10.5.x ou mais recente têm uma configuração padrão razoável, de modo que nenhum ajuste de servidor deve ser necessário e tudo o que é necessário pode ser configurado no lado do cliente.

Abaixo está um exemplo `/etc/my.cnf.d/server.cnf` para um servidor com 8 GB de RAM. Essas configurações devem ser suficientes para a maioria das instalações. MySQL e MariaDB têm ajustáveis que irão aumentar o desempenho do seu servidor que são considerados desnecessários, a menos que esteja a planejar ter uma grande quantidade de utilizadores simultâneos a acessar o sistema. Consulte a documentação de vários fornecedores sobre esses detalhes.

É absolutamente crítico para reduzir problemas durante a instalação que a configuração `innodb_file_per_table` esteja configurada corretamente e o MySQL/MariaDB reiniciado antes de iniciar a instalação do Weblate.

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

datadir=/var/lib/mysql

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

innodb_large_prefix=1
innodb_file_format=Barracuda
innodb_file_per_table=1
innodb_buffer_pool_size=2G
sql_mode=STRICT_TRANS_TABLES
```

Dica: No caso de estar a receber o erro #1071 – Specified key was too long; max key length is 767 bytes, atualize a sua configuração para incluir as configurações `innodb` acima e reinicie a sua instalação.

Dica: No caso de estar a receber o erro #2006 – MySQL server has gone away, configurar `CONN_MAX_AGE` pode ajudar.

Configurando Weblate para usar MySQL/MariaDB

O trecho de `settings.py` para MySQL e 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": {},
```

(continues on next page)

(continuação da página anterior)

```
}  
}
```

Também deve criar a conta de utilizador `weblate` no MySQL ou MariaDB antes de iniciar a instalação. Use os comandos abaixo para fazer isso:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';  
FLUSH PRIVILEGES;
```

2.1.6 Outras configurações

Configuração de e-mail de saída

O Weblate envia e-mails em várias ocasiões - para a ativação de contas e sobre várias notificações configuradas pelos utilizadores. Para isso, precisa de acesso a um servidor de SMTP.

A configuração do servidor de e-mail é configurada a usar essas configurações: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_SSL`, `EMAIL_USE_TLS`, `EMAIL_HOST_USER` e `EMAIL_PORT`. Os nomes deles são bastante autoexplicativos, mas pode encontrar mais informações na documentação do Django.

Dica: Caso tenha erro sobre autenticação não suportada (por exemplo, `SMTP AUTH extension not supported by server`), é provavelmente causada pelo uso de conexão insegura e o servidor se recusa a autenticar dessa forma. Tente ativar `EMAIL_USE_TLS` nesse caso.

Veja também:

Não a receber e-mails do Weblate, Configurando e-mail de saída em contentor Docker

Executar por trás de um proxy reverso

Vários recursos no Weblate dependem de ser capaz de obter o endereço IP do cliente. Isso inclui *Limitação de taxa*, *Proteção contra spam* ou *Registo de auditoria*.

Na configuração predefinida, o Weblate analisa o endereço IP de `REMOTE_ADDR` que é definido pelo manipulador WSGI.

Se estiver a usar um proxy reverso, este campo provavelmente conterá o seu endereço. Precisa configurar o Weblate para confiar em cabeçalhos HTTP adicionais e analisar o endereço IP destes. Isso não pode ser ativado por predefinição, porque permitiria a falsificação de endereços IP para instalações que não usam um proxy reverso. Ativar `IP_BEHIND_REVERSE_PROXY` pode ser suficiente para as configurações mais usuais, mas podia precisar de ajustar `IP_PROXY_HEADER` e `IP_PROXY_OFFSET` também.

Veja também:

Proteção contra spam, Limitação de taxa, Registo de auditoria, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER, IP_PROXY_OFFSET, SECURE_PROXY_SSL_HEADER

Proxy HTTP

O Weblate executa comandos VCS e esses que aceitam a configuração proxy do ambiente. A abordagem recomendada é definir configurações de proxy em `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Veja também:

Variáveis de Ambiente de Proxy

2.1.7 Ajustar a configuração

Veja também:

Amostra de configuração

Copie `weblate/settings_example.py` para `weblate/settings.py` e ajuste-o para corresponder à configuração. Provavelmente irá ajustar as opções a seguir: `ADMINS`

Lista de administradores de sites para receber notificações quando algo dá errado, por exemplo, notificações em mensagens fracassadas ou erros de Django.

Veja também:

`ADMINS`

`ALLOWED_HOSTS`

Precisa definir isso para listar os hosts que o seu site deve servir. Por exemplo:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternativamente, pode incluir curinga:

```
ALLOWED_HOSTS = ["*"]
```

Veja também:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Configuração de hosts permitidos*

`SESSION_ENGINE`

Configure como as suas sessões serão armazenadas. Caso mantenha o mecanismo de backend do banco de dados predefinido, deve agendar: **`weblate clearsessions`** para remover dados de sessão obsoletos do banco de dados.

Se estiver a usar o Redis como cache (veja *Ativar o cache*) é recomendado também usá-lo para sessões:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Veja também:

Configuring the session engine, `SESSION_ENGINE`

`DATABASES`

Conetividade ao servidor de banco de dados, verifique a documentação do Django para obter mais detalhes.

Veja também:

Configuração de banco de dados para o Weblate, `DATABASES`, *Databases*

DEBUG

Desative isto para qualquer servidor de produção. Com o modo de depuração ativado, o Django mostrará backtraces em caso de erro aos utilizadores, quando desativá-lo, erros serão enviados por e-mail para ADMINS (veja acima).

O modo de depuração também desacelera o Weblate, já que o Django armazena muito mais informações internamente neste caso.

Veja também:

`DEBUG`

DEFAULT_FROM_EMAIL

Endereço de remetente de e-mail para e-mail de saída, por exemplo, e-mails de registo.

Veja também:

`DEFAULT_FROM_EMAIL`

SECRET_KEY

Chave usada por Django para assinar informações em cookies, consulte *Chave secreta do Django* para obter mais informações.

Veja também:

`SECRET_KEY`

SERVER_EMAIL

E-mail usado como endereço de remetente para envio de e-mails ao administrador, por exemplo, notificações em mensagens falhadas.

Veja também:

`SERVER_EMAIL`

2.1.8 Preencher o banco de dados

Depois que a sua configuração estiver pronta, pode executar `weblate migrate` para criar a estrutura do banco de dados. Agora deve ser capaz de criar projetos de tradução a usar a interface administrativa.

Caso queira executar uma instalação não interativamente, pode usar `weblate migrate --noinput` e depois criar um utilizador administrativo pelo comando `createadmin`.

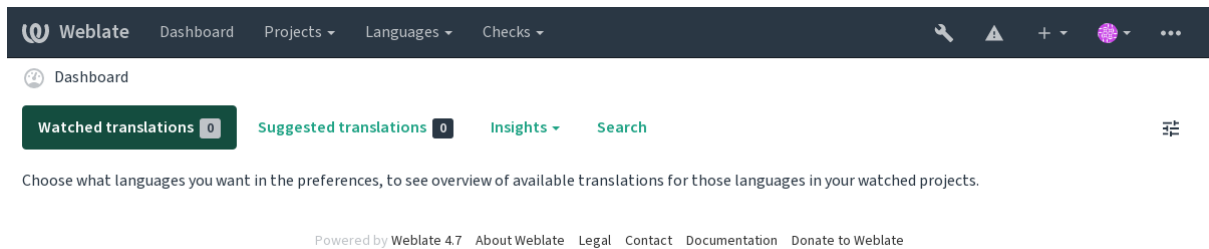
Uma vez feito, também deve verificar o *Relatório de desempenho* na interface administrativa, o que lhe dará dicas de configuração potencial não ideal no seu site.

Veja também:

Configuração, Lista de privilégios

2.1.9 Configuração de produção

Para uma configuração de produção, deve realizar ajustes descritos nas seções a seguir. As configurações mais críticas acionarão um aviso, que é indicado por um ponto de exclamação na barra superior se estiver conectado como um superutilizador:



Também é recomendado inspecionar verificações desencadeadas por Django (embora possa não precisar corrigir todas):

```
weblate check --deploy
```

Também pode revisar a mesma lista de verificação na *Interface de gestão*.

Veja também:

[Deployment checklist](#)

Desativar o modo de depuração

Desative o modo de depuração do Django (*DEBUG*) com:

```
DEBUG = False
```

Com o modo de depuração ativado, o Django armazena todas as consultas executadas e mostra aos utilizadores backtraces de erros, o que não é desejado numa configuração de produção.

Veja também:

[Ajustar a configuração](#)

Configurar administradores corretamente

Defina os endereços de administração corretos à configuração *ADMINS* para definir quem receberá e-mails caso algo dê errado no servidor, por exemplo:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Veja também:

[Ajustar a configuração](#)

Definir domínio correto do site

Ajuste o nome e o domínio do site na interface administrativa, caso contrário, ligações no RSS ou e-mails de registo não funcionarão. Isto é configurado usando *SITE_DOMAIN* que deve conter o nome de domínio do site.

Alterado na versão 4.2: Antes da versão 4.2, a estrutura de sites do Django era usada em vez disso, consulte [The “sites” framework](#).

Veja também:

[Configuração de hosts permitidos](#), [Configurar HTTPS corretamente](#) *SITE_DOMAIN*, *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

Configurar HTTPS corretamente

É fortemente recomendado executar Weblate a com o protocolo criptografado HTTPS. Depois de ativá-lo, deve definir `ENABLE_HTTPS` nas configurações:

```
ENABLE_HTTPS = True
```

Dica: Pode também configurar o HSTS, consulte [SSL/HTTPS](#) para obter mais detalhes.

Veja também:

[ENABLE_HTTPS](#), [Configuração de hosts permitidos](#), [Definir domínio correto do site](#)

Definir SECURE_HSTS_SECONDS corretamente

Se o seu site for servido sobre SSL, deve considerar definir um valor para `:django:`SECURE_HSTS_SECONDS`` no `settings.py` para ativar HTTP Strict Transport Security. Por padrão, ele está definido para 0 como mostrado abaixo.

```
SECURE_HSTS_SECONDS = 0
```

Se for definido como um valor inteiro não-zero, o cabeçalho `django.middleware.security.SecurityMiddleware` define o cabeçalho HTTP Strict Transport Security em todas as respostas que ainda não o possuem.

Aviso: Definir isto incorretamente pode quebrar irreversivelmente (por algum tempo) o seu site. Leia primeiro a documentação [HTTP Strict Transport Security](#).

Usar um poderoso mecanismo de banco de dados

- Por favor, use PostgreSQL para um ambiente de produção, consulte *[Configuração de banco de dados para o Weblate](#)* para obter mais informações.
- Use adjacent location for running the database server, otherwise the networking performance or reliability might ruin your Weblate experience.
- Check the database server performance or tweak its configuration, for example using [PGTune](#).

Veja também:

[Configuração de banco de dados para o Weblate](#), [Migrando de outros bancos de dados para o PostgreSQL](#), [Ajustar a configuração](#), [Databases](#)

Ativar o cache

Se for possível, use Redis do Django e ajuste a variável de configuração `CACHES`, por exemplo:

```
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",
```

(continues on next page)

(continuação da página anterior)

```

        "PARSER_CLASS": "redis.connection.HiredisParser",
    },
}
}

```

Dica: Caso altere as configurações do Redis para o cache, pode ser necessário ajustá-las também para o Celery, consulte *Tarefas de fundo a usar o Celery*.

Veja também:

Cache de avatares, Django's cache framework

Cache de avatares

Além do cache de Django, Weblate realiza cache de avatares. Recomenda-se usar um cache separado, baseado em ficheiros para este fim:

```

CACHES = {
    "default": {
        # Default caching backend setup, see above
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "unix:///var/run/redis/redis.sock?db=0",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 604800,
        "OPTIONS": {
            "MAX_ENTRIES": 1000,
        },
    },
}

```

Veja também:

ENABLE_AVATARS, *AVATAR_URL_PREFIX*, *Avatars*, *Ativar o cache*, Django's cache framework

Configurar envio de e-mail

O Weblate precisa enviar e-mails em várias ocasiões e esses e-mails devem ter um endereço de remetente correto, por favor, configure **:configuração:SERVER_EMAIL** e *DEFAULT_FROM_EMAIL* para combinar com o seu ambiente, por exemplo:

```

SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"

```

Nota: Para desativar o envio de e-mails pelo Weblate, defina *EMAIL_BACKEND* a `django.core.mail.backends.dummy.EmailBackend`.

Isso desativará *toda* a entrega de e-mail, incluindo e-mails de registo ou redefinição de palavra-passe.

Veja também:

Ajustar a configuração, Configuração de e-mail de saída, EMAIL_BACKEND, DEFAULT_FROM_EMAIL, SERVER_EMAIL

Configuração de hosts permitidos

Django requer `ALLOWED_HOSTS` para manter uma lista de nomes de domínio que o seu site pode servir, deixá-lo vazio bloqueará todas solicitações.

Caso isso não esteja configurado para corresponder ao seu servidor HTTP, terá erros como `Invalid HTTP_HOST header: '1.1.1.1'`. Pode ter que adicionar `'1.1.1.1'` ao `ALLOWED_HOSTS`.

Dica: No contentor Docker, isso está disponível como `WEBLATE_ALLOWED_HOSTS`.

Veja também:

ALLOWED_HOSTS, WEBLATE_ALLOWED_HOSTS, Definir domínio correto do site

Chave secreta do Django

A configuração `SECRET_KEY` é usada pelo Django para assinar cookies e você deve realmente gerar o seu próprio valor em vez de usar o da configuração do exemplo.

Pode gerar uma nova chave por `weblate/examples/generate-secret-key`, que vem com o Weblate.

Veja também:

SECRET_KEY

Directório inicial

Alterado na versão 2.1: Isso não é mais necessário, agora o Weblate armazena todos os seus dados em `DATA_DIR`.

O diretório home do utilizador que executa o Weblate deve existir e ser gravável por este utilizador. Isso é especialmente necessário se quiser usar o SSH para acessar repositórios privados, mas o Git pode precisar acessar este diretório também (dependendo da versão git que usa).

Pode alterar o diretório usado pelo Weblate em `settings.py`, por exemplo, para defini-lo como diretório `configuration` na árvore do Weblate:

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

Nota: No Linux e em outros sistemas como UNIX, o caminho ao diretório home do utilizador é definido em `/etc/passwd`. Muitas distribuições usam um diretório sem permissão de escrita como predefinição para utilizadores para servir conteúdo web (como `apache`, `www-data` ou `wwwrun`), então tem que executar o Weblate sob um utilizador diferente ou alterar essa configuração.

Veja também:

Acessando repositórios

Carregar modelos

Recomenda-se usar um carregador de modelo em cache para Django. Armazena modelos analisados e evita a necessidade de analisar cada solicitação. Pode configurá-lo a usar o trecho a seguir (a configuração `loaders` é importante aqui):

```
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",
                    ],
                ),
            ],
        },
    ],
]
```

Veja também:

`django.template.loaders.cached.Loader`

Executar tarefas de manutenção

Para um desempenho ideal, é uma boa ideia executar algumas tarefas de manutenção em segundo plano. Isso agora é feito automaticamente por *Tarefas de fundo a usar o Celery* e abrange as seguintes tarefas:

- Verificação de saúde da configuração (de hora em hora).
- Realização de commits de alterações pendentes (de hora em hora), consulte *Commits adiados* e *commit_pending*.
- Atualização de alertas de componentes (dialy).
- Atualização dos ramos remotos (nightly), consulte *AUTO_UPDATE*.
- Backup de memória de tradução para JSON (diariamente), consulte *dump_memory*.
- Tarefas de manutenção de texto completo e banco de dados (tarefas diárias e semanais), consulte *cleanup-trans*.

Alterado na versão 3.2: Desde a versão 3.2, a maneira predefinida de executar essas tarefas é usar o Celery e o Weblate já vem com a configuração adequada, consulte *Tarefas de fundo a usar o Celery*.

Codificação e localidades do sistema

As localidades do sistema devem ser configuradas para UTF-8. Na maioria das distribuições Linux, esta é a configuração predefinida. Se não é o caso no seu sistema, altere as localidades para a variante UTF-8.

Por exemplo, a editar `/etc/default/locale` e a definir lá `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'
```

Usar uma autoridade certificadora personalizada

O Weblate verifica os certificados SSL durante as solicitações HTTP. Caso esteja a usar uma autoridade de certificação personalizada que não seja confiável em maços predefinidos, terá que adicionar o seu certificado como confiável.

A abordagem preferida é fazer isso no nível do sistema. Consulte a documentação da sua distro para mais detalhes (por exemplo, no Debian isso pode ser feito a por o certificado da AC em `/usr/local/share/ca-certificates/` e executando `update-ca-certificates`).

Uma vez feito isso, as ferramentas do sistema confiarão no certificado e isso inclui o Git.

Para código em Python, precisará configurar solicitações para usar o pacote de AC do sistema em vez do fornecido. Isso pode ser conseguido pondo os seguintes trechos em `settings.py` (o caminho é específico do Debian):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Comprimir os ativos do cliente

O Weblate vem com um monte de ficheiros JavaScript e CSS. Por razões de desempenho, é bom comprimi-los antes de enviar para um cliente. Na configuração predefinida isso é feito rapidamente ao custo de pouca sobrecarga. Em grandes instalações, recomenda-se ativar o modo de compressão offline. Isso precisa ser feito na configuração e a compressão tem que ser acionada em cada atualização do Weblate.

A mudança da configuração é simples ao ativar `django.conf.settings.COMPRESS_OFFLINE` e configuração `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (este último já está incluído na configuração do exemplo):

```
COMPRESS_OFFLINE = True
```

Em cada implantação precisa compactar os ficheiros para corresponder à versão atual:

```
weblate compress
```

Dica: A imagem oficial do Docker já tem este recurso ativado.

Veja também:

[Common Deployment Scenarios](#), [Servir ficheiros estáticos](#)

2.1.10 Executar o servidor

Dica: No caso de não ter experiência com os serviços descritos abaixo, pode tentar seguir *Instalando a usar Docker*.

Precisará de vários serviços para executar o Weblate, a configuração recomendada consiste em:

- Servidor de banco de dados (consulte *Configuração de banco de dados para o Weblate*)
- Servidor de cache (consulte *Ativar o cache*)
- Servidor web frontend para ficheiros estáticos e terminação SSL (consulte *Servir ficheiros estáticos*)
- Servidor WSGI para conteúdo dinâmico (consulte *Configuração de amostra para NGINX e uWSGI*)
- Celery para executar tarefas em segundo plano (consulte *Tarefas de fundo a usar o Celery*)

Nota: Existem algumas dependências entre os serviços, por exemplo, o cache e o banco de dados devem estar em execução ao iniciar os processos de Celery ou uwsgi.

Na maioria dos casos, executará todos os serviços num único servidor (virtual), mas se a sua instalação estar muito carregada, pode dividir os serviços. A única limitação disso é que os servidores Celery e Wsgi precisam acessar `DATA_DIR`.

Nota: O processo de WSGI deve ser executado sob o mesmo utilizador que o processo do Celery, caso contrário, os ficheiros em `DATA_DIR` serão armazenados com propriedade mista, a levar a problemas de tempo de execução.

Veja também *Permissões do sistema de ficheiros* e *Tarefas de fundo a usar o Celery*.

Executar um servidor web

Executar o Weblate não é diferente de executar qualquer outro programa baseado em Django. Django é geralmente executado como uWSGI ou fcgi (consulte exemplos para diferentes servidores web abaixo).

Para fins de teste, pode usar o servidor web incorporado no Django:

```
weblate runserver
```

Aviso: NÃO USE ESTE SERVIDOR NUMA CONFIGURAÇÃO DE PRODUÇÃO. Não passou por auditorias de segurança ou testes de desempenho. Veja também a documentação de Django no `runserver`.

Dica: O servidor embutido do Django serve apenas ficheiros estáticos com `DEBUG` ativado, pois é destinado apenas ao desenvolvimento. Para uso em produção, consulte as configurações de wsgi em *Configuração de amostra para NGINX e uWSGI*, *Configuração de amostra para Apache*, *Configuração de amostra para Apache and Unicorn* e *Servir ficheiros estáticos*.

Servir ficheiros estáticos

Alterado na versão 2.4: Antes da versão 2.4, o Weblate não usava a estrutura de ficheiros estáticos do Django corretamente e a configuração era mais complexa.

Django precisa coletar os ficheiros estáticos dele num único diretório. Para isso, execute `weblate collectstatic --noinput`. Isso copiará os ficheiros estáticos num diretório especificado pela configuração `STATIC_ROOT` (isso é a predefinição para um diretório `static` dentro de `DATA_DIR`).

Recomenda-se servir ficheiros estáticos diretamente do seu servidor web. Deve usá-los para os seguintes caminhos:

/static/ Serve ficheiros estáticos para Weblate e a interface de administração (definida por `STATIC_ROOT`).

/media/ Usado para o envio de mídia pelo utilizador (por exemplo, capturas de ecrã).

/favicon.ico Deve ser reescrito para reescrever uma regra para servir `/static/favicon.ico`.

Veja também:

Configuração de amostra para NGINX e uWSGI, Configuração de amostra para Apache, Configuração de amostra para Apache and Gunicorn, Comprimir os ativos do cliente, Deploying Django, Deploying static files

Política de segurança de conteúdo

A configuração predefinida do Weblate ativa o middleware `weblate.middleware.SecurityMiddleware` que define cabeçalhos HTTP relacionados à segurança como `Content-Security-Policy` ou `X-XSS-Protection`. São configurados por predefinição para funcionar com o Weblate e a configuração dele, mas isso pode precisar de personalização no seu ambiente.

Veja também:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

Configuração de amostra para NGINX e uWSGI

Para executar o servidor web de produção, use o wrapper `wsgi` instalado com Weblate (no caso de ambiente virtual é instalado como `~/weblate-env/lib/python3.7/site-packages/weblate/wsgi.py`). Também não se esqueça de definir o caminho de pesquisa Python para o seu virtualenv (por exemplo, usando `virtualenv = /home/user/weblate-env` no uWSGI).

A configuração a seguir executa o Weblate como uWSGI sob o servidor web NGINX.

A configuração para NGINX (também disponível como `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/;
```

(continues on next page)

(continuação da página anterior)

```

    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;
}
}

```

Configuração para uWSGI (também disponível como `weblate/examples/weblate.uwsgi.ini`):

```

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

# 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

```

(continues on next page)

```
# 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
```

Veja também:

How to use Django with uWSGI

Configuração de amostra para Apache

Recomenda-se o uso de prefork de MPM ao usar WSGI com Weblate.

A configuração a seguir executa o Weblate como WSGI. Precisa ter ativado `mod_wsgi` (disponível como `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.
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

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

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

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

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

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

(continues on next page)

(continuação da página anterior)

```

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

</VirtualHost>

```

Nota: Weblate precisa do Python 3, então, por favor se certifique que está a executar a variante do Python 3 do modwsgi. Usualmente, está disponível como um pacote separado, por exemplo `libapache2-mod-wsgi-py3`.

Veja também:

Codificação e localidades do sistema, [How to use Django with Apache and mod_wsgi](#)

Configuração de amostra para Apache and Gunicorn

A configuração seguinte executa o Weblate em Gunicorn and Apache 2.4 (disponível como `weblate/examples/apache.gunicorn.conf`):

```

#
# VirtualHost for Weblate using gunicorn on localhost:8000
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
#>env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your
#>setup.
#
<VirtualHost *:443>
  ServerAdmin admin@weblate.example.org
  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/

```

(continues on next page)

```
ProxyPreserveHost On
</VirtualHost>
```

Veja também:

How to use Django with Gunicorn

A executar o Weblate sob o caminho

Novo na versão 1.3.

Recomenda-se o uso de prefork de MPM ao usar WSGI com Weblate.

Uma configuração do Apache de amostra para servir o Weblate sob /weblate. Novamente, a usar mod_wsgi (também disponível como weblate/examples/apache-path.conf):

```
#
# VirtualHost for Weblate, running under /weblate path
#
# This example assumes Weblate is installed in virtualenv in /home/weblate/weblate-
# ↪env
# and DATA_DIR is set to /home/weblate/data, please adjust paths to match your ↪
# ↪setup.
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

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

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

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

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

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

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

Adicionalmente, irá ter de ajustar o weblate/settings.py:

```
URL_PREFIX = "/weblate"
```

2.1.11 Tarefas de fundo a usar o Celery

Novo na versão 3.2.

O Weblate usa Celery para processar tarefas em segundo plano. Uma configuração típica a usar redis como backend se parece com isso:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Veja também:

Configuração do broker Redis no Celery

Para o desenvolvimento, pode usar uma configuração ansiosa, que processa todas as tarefas no local, mas isso terá impacto no desempenho do Weblate:

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

Também deve iniciar o worker do Celery para processar as tarefas e iniciar tarefas programadas, isso pode ser feito diretamente na linha de comando (o que é mais útil ao depurar ou desenvolver):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Nota: O processo de Celery deve ser executado sob o mesmo utilizador que o Weblate e o processo do WSGI, caso contrário, os arquivos em *DATA_DIR* serão armazenados com propriedade mista, a levar a problemas de tempo de execução.

Veja também *Permissões do sistema de ficheiros* e *Executar o servidor*.

Executando Celery como serviço do sistema

Provavelmente vai querer executar o Celery como um daemon e que é coberto por *Daemonization*. Para as configurações Linux mais comuns a usar systemd, pode usar os ficheiros de exemplo disponibilizados na pasta *examples* listado abaixo.

Unidade do systemd a ser posta como `/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} \'
```

(continues on next page)

(continuação da página anterior)

```
-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
```

Configuração do ambiente a ser posta como `/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"

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

Configuração adicional para alternar os logs do Celery a usar **logrotate** a ser posta como `/etc/logrotate.d/celery`:

```
/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}
```

Tarefas periódicas a usar a batida do Celery

O Weblate vem com configuração embutida para tarefas programadas. No entanto, pode definir tarefas adicionais em `settings.py`. Por exemplo, veja [Commits adiados](#).

As tarefas deveriam ser executadas por daemons de batidas do Celery. Caso não esteja a funcionar corretamente, pode não estar a funcionar ou o seu banco de dados foi corrompido. Verifique os registos de inicialização do Celery nesse caso para descobrir a causa raiz.

Monitorar o estado do Celery

Pode usar `celery_queues` para ver o comprimento atual das filas de tarefas do Celery. Caso a fila fique muito longa, também terá erro de configuração na interface administrativa.

Aviso: Os erros do Celery são por padrão apenas conectados ao log do Celery e não são visíveis ao utilizador. Caso queira ter uma visão geral sobre tais falhas, recomenda-se ajustar a configuração para ir [Coletando relatórios de erros](#).

Veja também:

[Configuration and defaults](#), [Workers Guide](#), [Daemonization](#), [Monitoring and Management Guide](#), `celery_queues`

2.1.12 Monitorando o Weblate

O Weblate fornece a URL `/healthz/` a ser usada em verificações de saúde simples, por exemplo, usando Kubernetes. O contentor Docker tem verificação de saúde embutida usando esta URL.

Para monitorar as métricas do Weblate, pode usar o ponto final `GET /api/metrics/` da API.

Veja também:

[Plugin do Weblate para Munin](#)

2.1.13 Coletando relatórios de erros

Weblate, como qualquer outro software, pode falhar. Para coletar estados de falha úteis, recomendamos usar serviços de terceiros para coletar tais informações. Isso é especialmente útil no caso de falhas nas tarefas do Celery, que de outra forma só relatariam erro nos logs e não será notificado sobre eles. O Weblate tem suporte para os seguintes serviços:

Sentry

O Weblate possui suporte embutido para [Sentry](#). Para usá-lo, é suficiente definir `SENTRY_DSN` no `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```


Rollbar

O Weblate tem suporte embutido para [Rollbar](#). Para usá-lo, basta seguir instruções para o notificador de Rollbar para Python.

Em suma, precisa ajustar `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",
}
```

Todo o resto é integrado automaticamente, agora coletará erros do lado do servidor e do cliente.

2.1.14 Migrando Weblate para outro servidor

Migrar o Weblate para outro servidor deve ser muito fácil, porém armazena dados em poucos locais que deve migrar cuidadosamente. A melhor abordagem é parar o Weblate para a migração.

Migrando banco de dados

Dependendo do backend do seu banco de dados, pode ter várias opções para migrar o banco de dados. O mais simples é despejar o banco de dados num servidor e importá-lo no novo. Alternativamente, pode usar a replicação no caso do seu banco de dados ter suporte a isso.

A melhor abordagem é usar ferramentas nativas de banco de dados, pois geralmente são as mais eficazes (por exemplo, **mysqldump** ou **pg_dump**). Se quiser migrar entre diferentes bancos de dados, a única opção pode ser usar a gestão do Django para despejar e importar o banco de dados:

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

Migrando repositórios VCS

Os repositórios VCS armazenados em `DATA_DIR` também precisam ser migrados. Pode simplesmente copiá-los ou usar **rsync** para fazer a migração de forma mais eficaz.

Outras notas

Não se esqueça de mover outros serviços que o Weblate pode ter usado como Redis, tarefas do cron ou backends de autenticação personalizada.

2.2 Implantações de Weblate

O Weblate pode ser facilmente instalado na sua nuvem. Encontre um guia detalhado para sua plataforma:

- *Instalando a usar Docker*
- *Instalando no OpenShift*
- *Instalando no Kubernetes*

2.2.1 Implantações de terceiros para Weblate

Nota: As seguintes implantações não são desenvolvidas ou suportadas pela equipa Weblate. Partes da configuração podem variar do que está descrito nesta documentação.

Pilha Weblate para Bitnami

Bitnami fornece uma pilha Weblate para muitas plataformas em <<https://bitnami.com/stack/weblate>>. A configuração será ajustada durante a instalação, consulte <<https://bitnami.com/stack/weblate/README.txt>> para mais documentação.

Pacote de Weblate para Cloudron

Cloudron é uma plataforma para auto-hospedagem de aplicações da web. Weblate instalado com Cloudron será automaticamente atualizado. O pacote é mantido pela equipa Cloudron no seu *repositório de pacotes Weblate* <<https://git.cloudron.io/cloudron/weblate-app>> _.



Weblate no YunoHost

O projeto de hospedagem própria YunoHost fornece um pacote para Weblate. Uma vez que tenha a sua instalação YunoHost, pode instalar o Weblate como qualquer outra aplicação. Ele fornecerá uma pilha de trabalho completo com backup e restauração, mas ainda pode ter que editar seu ficheiro de configurações para usos específicos.

Pode usar a sua interface de administração ou este botão (vai levá-lo ao seu servidor):



Também é possível usar a interface da linha de comando:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Atualizando o Weblate

2.3.1 Atualizações de imagem Docker

A imagem Docker oficial (veja [Instalando a usar Docker](#)) possui todas as etapas de atualização integradas. Não há nenhuma etapa manual além de buscar a versão mais recente.

2.3.2 Instruções genéricas de atualização

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 [Version specific instructions](#) 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 [Fazer backup e mover o Weblate](#).

1. Stop wsgi and Celery processes. The upgrade can perform incompatible changes in the database, so it is always safer to avoid old processes running while upgrading.
2. Upgrade Weblate code.

For pip installs it can be achieved by:

```
pip install -U Weblate
```

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

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

3. Upgrade configuration file, refer to `settings_example.py` or [Version specific instructions](#) for needed steps.
4. Upgrade database structure:

```
weblate migrate --noinput
```

5. Collect updated static files (see [Executar o servidor](#) and [Servir ficheiros estáticos](#)):

```
weblate collectstatic --noinput
```

6. Compactar ficheiros JavaScript e CSS (opcional, veja [Comprimir os ativos do cliente](#)):

```
weblate compress
```

7. If you are running version from Git, you should also regenerate locale files every time you are upgrading. You can do this by invoking:

```
weblate compilemessages
```

8. Verify that your setup is sane (see also *Configuração de produção*):

```
weblate check --deploy
```

9. Restart Celery worker (see *Tarefas de fundo a usar o Celery*).

2.3.3 Version specific instructions

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.

Veja também:

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.

Veja também:

Upgrade from 3.11 to 4.0 in [Weblate 4.0 documentation](#)

Upgrade from 4.0 to 4.1

Please follow *Instruções genéricas de atualização* 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.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.1 to 4.2

Please follow *Instruções genéricas de atualização* 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.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.2 to 4.3

Please follow *Instruções genéricas de atualização* 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 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`.

Alterado na versão 4.3.1:

- The Celery configuration was changed to add `memory` queue. Please adjust your startup scripts and `CELERY_TASK_ROUTES` setting.

Alterado na versão 4.3.2:

- The `post_update` method of addons now takes extra `skip_push` parameter.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.3 to 4.4

Please follow *Instruções genéricas de atualização* in order to perform update.

Notable configuration or dependencies changes:

- There is a change in `INSTALLED_APPS`, `weblate.configuration` has to be added there.
- Django 3.1 is now required.
- In case you are using MySQL or MariaDB, the minimal required versions have increased, see *MySQL e MariaDB*.

Alterado na versão 4.4.1:

- *Gettext monolíngue* 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.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.4 to 4.5

Please follow *Instruções genéricas de atualização* in order to perform update.

Notable configuration or dependencies changes:

- The migration might take considerable time if you had big glossaries.
- Glossaries are now stored as regular components.
- The glossary API is removed, use regular translation API to access glossaries.
- There is a change in `INSTALLED_APPS` - `weblate.metrics` should be added.

Alterado na versão 4.5.1:

- There is a new dependency on the *pyahocorasick* module.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.5 to 4.6

Please follow *Instruções genéricas de atualização* 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`.
- A API para a criação de componentes agora usa automaticamente *URLs internas do Weblate*, veja `POST /api/projects/(string:project)/components/`.
- There is a change in dependencies and `PASSWORD_HASHERS` to prefer Argon2 for passwords hashing.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.6 to 4.7

Please follow *Instruções genéricas de atualização* 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.

Veja também:

Instruções genéricas de atualização

2.3.4 Upgrading from Python 2 to Python 3

Weblate no longer supports Python older than 3.5. In case you are still running on older version, please perform migration to Python 3 first on existing version and upgrade later. See [Upgrading from Python 2 to Python 3 in the Weblate 3.11.1 documentation](#).

2.3.5 Migrando de outros bancos de dados para o 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.

Criar um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate num banco de dados separado e separar a conta do utilizador:

```
# 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 *Migrating to PostgreSQL using pgloader*.

1. Add PostgreSQL as additional database connection to the `settings.py`:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
```

(continues on next page)

(continuação da página anterior)

```

    "USER": "weblate",
    # Database password
    "PASSWORD": "password",
    # Set to empty string for localhost
    "HOST": "database.example.com",
    # Set to empty string for default
    "PORT": "",
    # Additional database options
    "OPTIONS": {
        # In case of using an older MySQL server, which has MyISAM as a
        ↪default storage
        # 'init_command': 'SET storage_engine=INNODB',
        # Uncomment for MySQL older than 5.7:
        # 'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
        # If your server supports it, see the Unicode issues above
        "charset": "utf8mb4",
        # Change connection timeout in case you get MySQL gone away error:
        "connect_timeout": 28800,
    },
},
"postgresql": {
    # Database engine
    "ENGINE": "django.db.backends.postgresql",
    # Database name
    "NAME": "weblate",
    # Database user
    "USER": "weblate",
    # Database password
    "PASSWORD": "password",
    # Set to empty string for localhost
    "HOST": "database.example.com",
    # Set to empty string for default
    "PORT": "",
},
}

```

2. Run migrations and drop any data inserted into the tables:

```

weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql

```

3. Dump legacy database and import to PostgreSQL

```

weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql

```

4. Adjust DATABASES to use just PostgreSQL database as default, remove legacy connection.

Weblate should be now ready to run from the PostgreSQL database.

Migrating to PostgreSQL using pgloader

The [pgloader](#) is a generic migration tool to migrate data to PostgreSQL. You can use it to migrate Weblate database.

1. Adjust your `settings.py` to use PostgreSQL as a database.
2. Migrate the schema in the PostgreSQL database:

```
weblate migrate
weblate sqlflush | weblate dbshell
```

3. Run the `pgloader` to transfer the data. The following script can be used to migrate the database, but you might want to learn more about [pgloader](#) to understand what it does and tweak it to match your setup:

```
LOAD DATABASE
FROM      mysql://weblate:password@localhost/weblate
INTO      postgresql://weblate:password@localhost/weblate

WITH include no drop, truncate, create no tables, create no indexes, no_
↳foreign keys, disable triggers, reset sequences, data only

ALTER SCHEMA 'weblate' RENAME TO 'public'
;
```

2.3.6 Migrating from Pootle

As Weblate was originally written as replacement from Pootle, it is supported to migrate user accounts from Pootle. You can dump the users from Pootle and import them using `importusers`.

2.4 Fazer backup e mover o Weblate

2.4.1 Backup automatizado pelo BorgBackup

Novo na versão 3.9.

O Weblate tem suporte embutido para criação de backups de serviços a usar [BorgBackup](#). Borg cria backups criptografados eficazes em termos de espaço que podem ser armazenados com segurança na nuvem. Os backups podem ser controlados na interface de gestão da guia *Backups*.

Alterado na versão 4.4.1: Os bancos de dados PostgreSQL e MySQL/MariaDB estão incluídos nos backups automatizados.

Os backups que usam o Borg são incrementais e o Weblate é configurado para manter os seguintes backups:

- Backups diários para 14 dias
- Backups semanais por 8 semanas atrás
- Backups mensais por 6 meses atrás

[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/tmpf7o8joh_weblate

Backup service credentials June 16, 2021

Backup repository /tmp/tmpf7o8joh_weblate

Passphrase 4KbP5S3YUe53mRKEPHeEJg6KAf#Pb%svAiztrgr!jM\$Jv\$9q@m
 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 June 16, 2021

Backup performed June 16, 2021

Repository initialization June 16, 2021

[Turn off](#)
[Perform backup](#)
[Delete](#)

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

[Activate](#)
[Purchase support package](#)

Add backup service

Backup repository URL

Use /path/to/repo for local backups or user@host:/path/to/repo for remote SSH backups.

[Add](#)

Powered by Weblate 4.7
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

Chave de criptografia do Borg

BorgBackup cria backups criptografados e não conseguiria restaurá-los sem a palavra-passe. A palavra-passe é gerada ao adicionar novo serviço de backup e deve copiá-lo e mantê-lo num lugar seguro.

Se estiver a usar *Armazenamento de backup provisionado do Weblate*, faça backup da sua chave SSH privada também, pois ela é usada para acessar os seus backups.

Veja também:

`borg init`

2.4.2 Armazenamento de backup provisionado do Weblate

A forma mais fácil de fazer backup da sua instância do Weblate é comprar o [serviço de backup em weblate.org](https://weblate.org/support/#backup). É assim que o faz funcionar:

1. Compre o *Serviço de backup* em <https://weblate.org/support/#backup>.
2. Insira a chave obtida na interface de gestão, veja *Integrando o apoio*.
3. Weblate se conecta ao serviço de nuvem e obtém informações de acesso para os backups.
4. Ative a nova configuração de backup a partir da guia *Backups*.
5. Faça backup das suas credenciais do Borg para conseguir restaurar os backups, veja *Chave de criptografia do Borg*.

Dica: O passo manual para ativar tudo está lá para sua segurança. Sem o seu consentimento, nenhum dado é enviado ao repositório de backup obtido através do processo de registo.

2.4.3 Usar armazenamento de backup personalizado

Também pode usar o seu próprio armazenamento para backups. SSH pode ser usado para armazenar cópias de segurança no destino remoto, o servidor de destino precisa do **BorgBackup** instalado.

Veja também:

[General](#) na documentação do Borg

Sistema de ficheiros local

Recomenda-se especificar o caminho absoluto para o backup local, por exemplo `/caminho/para/backup`. O diretório deve poder ser escrito pelo utilizador a executar o Weblate (veja *Permissões do sistema de ficheiros*). Se ele não existir, o Weblate tenta criá-lo, mas precisa das permissões apropriadas para fazê-lo.

Dica: Ao executar o Weblate no Docker, certifique-se de que o local de backup seja exposto como um volume do contentor Weblate. Caso contrário, os backups serão descartados pelo Docker na reinicialização do seu contentor.

Uma opção é pôr backups num volume existente. Por exemplo, `/app/data/borgbackup`. Este é um volume existente no contentor.

Também pode adicionar um novo contentor para os backups no ficheiro de composição do Docker, por exemplo, a usar `/borgbackup`:

```
services:
  weblate:
    volumes:
```

(continues on next page)

(continuação da página anterior)

```
- /home/weblate/data:/app/data
- /home/weblate/borgbackup:/borgbackup
```

O diretório onde os backups serão armazenados para serem possuídos por UID 1000, caso o contrário, Weblate não será capaz de escrever os backups lá.

Backups remotos

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 [Chave SSH do Weblate](#)).
6. Configure the backup location in Weblate as `user@host:/path/to/backups`.

Dica: [Armazenamento de backup provisionado do Weblate](#) provides you automated remote backups without any effort.

Veja também:

[Chave SSH do Weblate](#)

2.4.4 Restaurar do BorgBackup

1. Restaurar o acesso ao repositório de backup e preparar a sua palavra-passe de backup.
2. Liste todos os backups no servidor a usar `borg list REPOSITÓRIO`.
3. Restaure o backup desejado para o diretório atual a usar `borg extract REPOSITÓRIO::PACOTE`.
4. Restaure o banco de dados do despejo de SQL posto no diretório `backup` no diretório de dados do Weblate (veja `:ref:"backup-dumps"`).
5. Copie a configuração do Weblate (`backups/settings.py`, veja [Dados despejados para backups](#)) até o local correto, veja [Ajustar a configuração](#).

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 [Dados despejados para backups](#)).

6. Copie todo o diretório de dados restaurados para o local configurado por `DATA_DIR`.

When using Docker container place the data into the data volume, see [Volumes de contentor Docker](#).

Please make sure the files have correct ownership and permissions, see [Permissões do sistema de ficheiros](#).

A sessão dos Borg pode parecer com isso:

```
$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
2019-09-26T14:56:08          Thu, 2019-09-26 14:56:08
→ [de0e0f13643635d5090e9896bdaceb92a023050749ad3f3350e788f1a65576a5]
```

(continues on next page)

(continuação da página anterior)

```
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:
```

Veja também:

`borg list`, `borg extract`

2.4.5 Backup manual

Dependendo do que deseja gravar, faça backup do tipo de dados que o Weblate armazena em cada lugar.

Dica: Se estiver a fazer os backups manualmente, pode silenciar os avisos do Weblate sobre a falta de backups a adicionar `weblate.I028` para `SILENCED_SYSTEM_CHECKS` em `settings.py` ou `WEBLATE_SILENCED_SYSTEM_CHECKS` para o Docker.

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

Banco de dados

O local de armazenamento real depende da configuração do seu banco de dados.

Dica: O banco de dados é o armazenamento mais importante. Configure backups regulares do seu banco de dados. Sem o banco de dados, todas as traduções são perdidas.

Backup nativo do banco de dados

A abordagem recomendada é gravar um despejo do banco de dados a usar ferramentas nativas, tais como `pg_dump` ou `mysqldump`. Esta abordagem normalmente tem um desempenho melhor do que o backup do Django e restaura tabelas completas com todos os seus dados.

Pode restaurar esse backup numa versão mais nova do Weblate, ele executará todas as migrações necessárias ao executar em `migrate`. Consulte [Atualizando o Weblate](#) sobre informações mais detalhadas sobre como a atualizar entre as versões.

Backup do banco de dados do Django

Alternativamente, pode fazer backup do seu banco de dados a utilizar o comando `dumpdata` do Django. Dessa forma o backup é agnóstico de banco de dados e pode ser usado caso queira alterar o backend do banco de dados.

Antes de restaurar o banco de dados, precisa estar a usar exatamente a mesma versão do Weblate na qual o backup foi feito. Isso é necessário, pois a estrutura do banco de dados muda entre as versões e acabaria a corromper os dados de alguma forma. Depois de instalar a mesma versão, execute todas as migrações do banco de dados a usar `migrate`.

Depois disso, algumas entradas já serão criadas no banco de dados e as terá no backup do banco de dados também. A abordagem recomendada é apagar essas entradas manualmente a usar o shell de gestão (veja [Invocando comandos de gestão](#)):

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

Ficheiros

Se tiver espaço de backup suficiente, basta fazer backup de todo o `DATA_DIR`. Esta é uma aposta segura, mesmo que inclua alguns ficheiro que não quer. As seções a seguir descrevem o que deve fazer backup e o que pode pular em detalhes.

Dados despejados para backups

Alterado na versão 4.7: The environment dump was added as `environment.yml` to help in restoring in the Docker environments.

Armazenados em `DATA_DIR/backups`.

O Weblate despeja vários dados aqui e pode incluir esses ficheiros para backups mais completos. Os ficheiros são atualizados diariamente (requer um servidor de «beats» do Celery em execução, consulte *Tarefas de fundo a usar o Celery*). Atualmente, isto inclui:

- Configurações do Weblate como `settings.py` (existe também a versão expandida em `settings-expanded.py`).
- Backup de banco de dados PostgreSQL como `database.sql`.
- Environment dump as `environment.yml`.

Os backups do banco de dados são salvos como texto simples por padrão, mas eles também podem ser comprimidos ou totalmente ignorados a usar `DATABASE_BACKUP`.

To restore the database backup load it using database tools, for example:

```
psql --file=database.sql weblate
```

Repositórios de controle de versão

Armazenados em `DATA_DIR/vcs`.

Os repositórios de controle de versão contêm uma cópia dos seus repositórios upstream com alterações do Weblate. Se tiver o *Enviar ao submeter* ativado para todos os seus componentes de tradução, todas as alterações do Weblate são incluídas no upstream. Não há necessidade de fazer backup dos repositórios no lado do Weblate, pois eles podem ser clonados novamente a partir dos upstream locais sem perda de dados.

Chaves SSH e GPG

Armazenados em `DATA_DIR/ssh` e `DATA_DIR/home`.

Se está a usar chaves SSH ou GPG geradas pelo Weblate, deve fazer backup destes locais. Caso contrário, vai perder as chaves privadas e terá que gerar novamente as novas.

Ficheiros enviados pelo utilizador

Armazenados em `DATA_DIR/media`.

Deve fazer o backup de todos os ficheiros enviados pelo utilizador (por exemplo, *Contexto visual para cadeias*).

Tarefas do Celery

A fila de tarefas do Celery pode conter algumas informações, mas geralmente não é necessária para um backup. No máximo, perderá atualizações ainda não processadas para a memória de tradução. Recomenda-se realizar a atualização de texto completo ou repositório ao restaurar de qualquer maneira, de modo que não há problema em perdê-las.

Veja também:

Tarefas de fundo a usar o Celery

Linha de comando para backup manual

Usando uma tarefa de cron, pode configurar um comando do Bash para ser executado diariamente, por exemplo:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz_
↪backups vcs ssh home media fonts secret
```

A cadeia entre as aspas após `XZ_OPT` permite que escolha as suas opções do xz, por exemplo, a quantidade de memória utilizada para compressão; veja <https://linux.die.net/man/1/xz>

Pode ajustar a lista de pastas e ficheiros às suas necessidades. Para evitar gravar a memória de tradução (na pasta backups), pode usar:

```
$ 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 backup manual

1. Restaure todos os dados dos quais tenha feito backup.
2. Atualize todos repositórios a usar o `updategit`.

```
weblate updategit --all
```

2.4.7 Mover uma instalação do Weblate

Realoque a instalação de um sistema diferente, a seguir as instruções de backup e restauração acima.

Veja também:

Upgrading from Python 2 to Python 3, Migrando de outros bancos de dados para o PostgreSQL

2.5 Autenticação

2.5.1 Registo de utilizador

A configuração predefinida para Weblate é usar python-social-auth, um formulário no site para lidar com o registo de novos utilizadores. Depois de confirmar o seu e-mail, um novo utilizador pode contribuir ou autenticar a usar um dos serviços de terceiros.

Também pode desativar o registo de novos utilizadores configurando `REGISTRATION_OPEN`.

As tentativas de autenticação estão sujeitas a *Limitação de taxa*.

2.5.2 Backends de autenticação

A solução embutida do Django é utilizada para autenticação, incluindo várias opções sociais para fazê-lo. Utilizando-a, pode importar o banco de dados de utilizadores de outros projetos baseados no Django (veja [Migrating from Pootle](#)).

Django pode, adicionalmente, ser configurado para autenticar em outros meios também.

Veja também:

[Configurações de autenticação](#) descreve como configurar a autenticação na imagem oficial do Docker.

2.5.3 Autenticação social

Graças ao [Welcome to Python Social Auth's documentation!](#), o Weblate tem suporte a autenticação utilizando muitos serviços de terceiros, tais como GitLab, Ubuntu, Fedora, etc.

Por favor, verifique a documentação deles por instruções de configuração genéricas em [Django Framework](#).

Nota: Por predefinição, o Weblate conta com serviços de autenticação de terceiros para fornecer um endereço de e-mail validado. Se alguns dos serviços que deseja usar não suportarem isto, por favor aplique a validação de e-mail no lado Weblate configurando `FORCE_EMAIL_VALIDATION` para eles. Por exemplo:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Veja também:

[Pipeline](#)

Permitir backends individuais é bastante fácil, é apenas uma questão de adicionar uma entrada à configuração `AUTHENTICATION_BACKENDS` e possivelmente adicionar chaves necessárias para um determinado método de autenticação. Por favor, note que alguns backends não fornecem e-mails do utilizador por predefinição, tem que solicitá-lo explicitamente, caso contrário o Weblate não será capaz de corretamente dar mérito às contribuições que os utilizadores fazem.

Dica: Most of the authentication backends require HTTPS. Once HTTPS is enabled in your web server please configure Weblate to report it properly using [ENABLE_HTTPS](#), or by [WEBLATE_ENABLE_HTTPS](#) in the Docker container.

Veja também:

[Backend de Python Social Auth](#)

Autenticação por OpenID

Para serviços baseados em OpenID, geralmente é apenas uma questão de ativá-los. A secção a seguir permite a autenticação OpenID para OpenSUSE, Fedora e 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",
)
```

Veja também:

[OpenID](#)

Autenticação por GitHub

Precisa registrar uma aplicação de OAuth no GitHub e, em seguida, dizer ao Weblate todos os seus segredos:

```
# 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"]
```

O GitHub deve ser configurado para ter URL de um retorno de chamada como `https://example.com/accounts/complete/github/`.

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte [Definir domínio correto do site](#).

Veja também:

[GitHub](#)

Autenticação por Bitbucket

Precisa registrar uma aplicação no Bitbucket e dar todos os segredos dele ao Weblate:

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

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

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte [Definir domínio correto do site](#).

Veja também:

[Bitbucket](#)

OAuth 2 do Google

Para usar o OAuth 2 do Google, precisa registrar-se numa aplicação em <<https://console.developers.google.com/>> e ativar a API do Google+.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/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: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

[Google](#)

OAuth 2 do Facebook

Como de costume com os serviços OAuth 2, precisa registrar a sua aplicação no Facebook. Uma vez feito, pode configurar o Weblate para usá-lo:

A URL de redirecionamento é `https://SERVIDOR WEBLATE/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: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

[Facebook](#)

OAuth 2 do GitLab

Para usar o OAuth 2 do GitLab, precisa registrar uma aplicação em <<https://gitlab.com/profile/applications>>.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/gitlab/` e garantir que marque o escopo `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: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

GitLab

Active Directory do Microsoft Azure

Weblate pode ser configurado para usar inquilinos comuns ou específicos para autenticação.

O URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/azuread-oauth2/` para autenticação comum e `https://SERVIDOR WEBLATE/accounts/complete/azuread-tenant-oauth2/` para autenticação específica do inquilino.

```
# Azure AD common

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

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

```
# Azure AD Tenant

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

# OAuth2 keys
```

(continues on next page)

(continuação da página anterior)

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

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

Microsoft Azure Active Directory

Slack

Para usar o OAuth 2 do Slack, precisa registrar uma aplicação em <<https://api.slack.com/apps>>.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/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: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

Slack

Overriding authentication method names and icons

You can override the authentication method display name and icon using settings as `SOCIAL_AUTH_<NAME>_IMAGE` and `SOCIAL_AUTH_<NAME>_TITLE`. For example overriding naming for Auth0 would look like:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Desativar autenticação por palavra-passe

Autenticação por e-mail e palavra-passe pode ser desativada através da remoção de `social_core.backends.email.EmailAuth` de `AUTHENTICATION_BACKENDS`. Mantenha sempre `weblate.accounts.auth.WeblateUserBackend` lá, pois é necessário para a funcionalidade central do Weblate.

Dica: Ainda pode usar autenticação por palavra-passe para a interface administrativa, para utilizadores que lá cria manualmente. Basta navegar para `/admin/`.

Por exemplo, a autenticação a usar apenas o provedor Open ID do openSUSE pode ser alcançada a usar o seguinte:

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

2.5.4 Autenticação por palavra-passe

A predefinição `settings.py` vem com um razoável conjunto de `AUTH_PASSWORD_VALIDATORS`:

- As palavras-passe não podem ser muito similares com as suas outras informações pessoais.
- As palavras-passe devem conter no mínimo de 10 caracteres.
- As palavras-passe não podem ser palavras-passe comumente usadas.
- As palavras-passe não podem ser inteiramente numéricas.
- As palavras-passe não podem consistir num único caractere ou apenas espaço em branco.
- As palavras-passe não podem corresponder a uma palavra-passe que já usou no passado.

Pode personalizar esta configuração para corresponder à sua política de palavra-passe.

Além disso, também pode instalar o `django-zxcvbn-password` o que dá bastante estimativas realistas de complexidade da palavra-passe e permite rejeitar palavras-passe abaixo de um determinado limite.

2.5.5 Autenticação por SAML

Novo na versão 4.1.1.

Siga as instruções do Python Social Auth para configuração. Diferenças notáveis:

- Weblate tem suporte a único IDP que tem de ser chamado de `weblate` em `SOCIAL_AUTH_SAML_ENABLED_IDPS`.
- A URL de metadados XML de SAML é `/accounts/metadata/saml/`.
- As configurações a seguir são preenchidas automaticamente: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Exemplo de configuração:

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

# Social auth backends setup
```

(continues on next page)

(continuação da página anterior)

```

SOCIAL_AUTH_SAML_SP_ENTITY_ID = f"https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbn1+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}

```

The default configuration extracts user details from following attributes, configure your IDP to provide them:

Attribute	SAML URI reference
Nome completo	urn:oid:2.5.4.3
Primeiro nome	urn:oid:2.5.4.42
Last name	urn:oid:2.5.4.4
E-mail	urn:oid:0.9.2342.19200300.100.1.3
Nome de utilizador	urn:oid:0.9.2342.19200300.100.1.1

Dica: The example above and the Docker image define an IDP labelled `weblate`. You might need to configure this string as *Relay* in your IDP.

Veja também:

Configurando SAML no Docker, SAML

2.5.6 Autenticação por LDAP

A autenticação por LDAP pode ser melhor alcançada utilizando o pacote *django-auth-ldap*. Pode instalá-lo através dos meios habituais:

```

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

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

```

Dica: This package is included in the Docker container, see *Instalando a usar Docker*.

Nota: Há algumas incompatibilidades no módulo Python LDAP 3.1.0, o que o pode impedir de usar essa versão. Se obter o erro `AttributeError: "module" object has no attribute "_trace_level"`, fazendo o downgrade para python-ldap 3.0.0 pode ajudar.

Uma vez que tenha o pacote instalado, pode ligá-lo à autenticação do 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: Deve remover `'social_core.backends.email.EmailAuth'` da configuração `AUTHENTICATION_BACKENDS`, caso contrário, os utilizadores poderão definir a palavra-passe deles no Weblate e autenticar a usar-a. Manter `'weblate.accounts.auth.WeblateUserBackend'` ainda é necessário para fazer permissões e facilitar utilizadores anónimos. Também permitirá que faça login a usar uma conta administrativa local, se a criou (por exemplo, a usar `createadmin`).

Usando palavra-passe associada

Se não puder usar a associação direta para autenticação, precisará usar a pesquisa e fornecer um utilizador para associar à pesquisa. Por exemplo:

```
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)"
)
```

Integração com o 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
```

Veja também:

[Django Authentication Using LDAP, Authentication](#)

2.5.7 Autenticação por CAS

A autenticação por CAS pode ser alcançada a usar um pacote como o *django-cas-ng*.

O primeiro passo é divulgar o campo de e-mail do utilizador via CAS. Isso tem que ser configurado no próprio servidor CAS e requer que utilize pelo menos CAS v2, já que o CAS v1 não tem suporte a atributos.

O segundo passo é atualizar a Weblate para utilizar o seu servidor CAS e os seus atributos.

Para instalar *django-cas-ng*:

```
pip install django-cas-ng
```


Uma vez que o pacote está instalado, pode conectá-lo ao sistema de autenticação do Django a modificar o ficheiro `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, um sinal pode ser usado para mapear o campo de e-mail para o objeto do utilizador. Para que isso funcione, tem que importar o sinal do pacote *django-cas-ng* e conectar o seu código com este sinal. Fazer isto em configurações de ficheiro pode causar problemas, portanto, é sugerido pô-lo:

- No método `django.apps.AppConfig.ready()` da configuração do seu app
- No ficheiro `urls.py` do projeto (quando não há 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()
```

Veja também:

Django CAS NG

2.5.8 Configurando autenticação por Django de terceiros

Geralmente, qualquer extensão de autenticação Django deve funcionar com Weblate. Basta seguir as instruções da extensão, lembrando de manter o backend do utilizador Weblate instalado.

Veja também:

Autenticação por LDAP, Autenticação por CAS

Normalmente, a instalação consiste em adicionar uma autenticação de backend a `AUTHENTICATION_BACKENDS` e a instalar uma app de autenticação (se houver) no `:setting: django:INSTALLED_APPS`:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Controlo de acesso

O Weblate vem com um sistema de privilégios fino para atribuir permissões ao utilizador para toda a instância ou num escopo limitado.

Alterado na versão 3.0: Antes do Weblate 3.0, o sistema de privilégios era baseado apenas no sistema de privilégios do Django, mas é agora especificamente construído para Weblate. Se for usado algo mais antigo, consulte a documentação para a versão específica que está a usar.

2.6.1 Controle de acesso personalizado

Se não está a administrar toda a instalação do Weblate e apenas tem acesso para gerir certos projetos (como no [Hosted Weblate](#)), as suas opções de gestão de controle de acesso são limitadas às seguintes configurações. Se não precisa de nenhuma configuração complexa, elas são suficientes para si.

Controlo de acesso ao projeto

Nota: Este recurso está indisponível para os projetos que executam o plano Libre no Hosted Weblate.

Pode limitar o acesso do utilizador a projetos individuais a seleccionar uma configuração diferente de *Controle de acesso*. As opções disponíveis são:

Pública Visível publicamente, traduzível para todos os utilizadores autenticados.

Protegido Publicamente visível, mas somente traduzível por utilizadores seleccionados.

Privado Visível e traduzível apenas por utilizadores seleccionados.

Personalizado Os recursos de *Gestão de utilizadores* serão desativados; por padrão, todos os utilizadores estão proibidos de realizar qualquer ação no projeto. Terá que configurar todas as permissões a usar *Controle de acesso personalizado*.

O *Controle de acesso* pode ser alterado na guia *Acesso* da configuração (*Gerir* ↓ *Configurações*) de cada projeto.

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 / Settings' is shown. The 'Access' tab is active, with other tabs being 'Basic', 'Workflow', and 'Components'. Under 'Access control', there are four radio button options: 'Public' (selected), 'Protected', 'Private', and 'Custom'. 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 4.7' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

O valor padrão pode ser alterado por `DEFAULT_ACCESS_CONTROL`.

Nota: Mesmo para projetos *Privados*, algumas informações sobre o seu projeto serão expostas: estatísticas e resumo de linguagem para toda a instância incluirão contagens para todos os projetos, apesar da configuração de controle de acesso. O nome do seu projeto e outras informações não podem ser revelados por meio disso.

Nota: O conjunto real de permissões disponíveis para utilizadores por padrão em projetos *Públicos*, *Protegidos* e *Privados* pode ser redefinido pelo administrador da instância de Weblate a usar *configurações personalizadas*.

Aviso: Ao ativar o controle de acesso *Personalizado*, o Weblate removerá todos os *grupos especiais* que ele criou para um projeto selecionado. Se estiver a fazer isso sem permissão administrativa para toda a instância de Weblate, perderá instantaneamente o seu acesso para gerir o projeto.

Veja também:

Controlo de acesso

A gerir controle de acesso por projeto

Os utilizadores com o privilégio *Gerir acesso ao projeto* (veja a *Lista de privilégios*) podem gerir utilizadores em projetos com controle de acesso não *Personalizado*. Eles podem atribuir os utilizadores a um dos seguintes grupos.

Para projetos *Públicos*, *Protegidos* e *Privados*:

Administração Inclui todas as permissões disponíveis para o projeto.

Revisar (somente se *fluxo de trabalho de revisão* estiver ativado) Pode aprovar traduções durante a revisão.

Para projetos *Protegidos* e *Privados* apenas:

Traduzir Pode traduzir o projeto e enviar traduções feitas offline.

Fontes Pode editar cadeias fonte (se permitido nas *configurações do projeto*) e informações de cadeias fonte.

Idiomas Pode gerir idiomas traduzidos (adicionar ou remover traduções).

Glossário Pode gerir glossário (adicionar ou remover entradas e também enviar).

Memória Pode gerir memória de tradução.


Capturas de ecrã Pode gerir capturas de ecrã (adicioná-las ou removê-las e associá-las a cadeias fonte).


VCS Pode gerir VCS e acessar o repositório exportado.

Faturação Pode acessar informações de cobrança e configurações (consulte *Faturação*).

Infelizmente, não é possível alterar este conjunto predefinido de grupos por enquanto. Além disso, dessa forma não é possível dar apenas algumas permissões adicionais a todos os utilizadores.

Nota: Para controle de acesso não-*Customizado*, uma instância de cada grupo descrito acima é na verdade definida para cada projeto. O nome real destes grupos será `Projeto@Grupo`, também exibido na interface de administração do Django desta forma. Embora eles não possam ser editados a partir da interface de utilizador do Weblate.

 Weblate
 Dashboard Projects Languages Checks

 WeblateOrg / Access control

Users API access

Users

Username	Full name	E-mail	Last login	Administration	Billing	Glossary	Languages	Memory	Screenshots	Sources	Translate	VCS
testuser	Weblate Test	weblate@example.org	18 seconds ago	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

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 4.7 About Weblate Legal Contact Documentation Donate to Weblate

Esses recursos estão disponíveis na página *Controle de acesso*, que pode ser acessada no projeto a partir do menu *Gerir* ↓ *Utilizadores*.

Novo convite de utilizador

Além de adicionar um utilizador existente ao projeto, também é possível convidar novos. Qualquer novo utilizador será criado imediatamente, mas a conta permanecerá inativa até que entre com uma ligação no convite enviado por e-mail. Não é necessário ter nenhum privilégio de todo o site para fazer isso, a permissão de gestão de acesso no escopo do projeto (por exemplo, uma associação no grupo *Administração*) seria suficiente.

Dica: Se o utilizador convidado perdeu a validade do convite, ele pode definir a sua palavra-passe a usar o endereço de e-mail convidado no formulário de redefinição de palavra-passe, pois a conta já foi criada.

Novo na versão 3.11: É possível reenviar o e-mail para convites de utilizador (a invalidar qualquer convite enviado anteriormente).

O mesmo tipo de convite está disponível em todo o site a partir da *interface de gestão* na guia *Utilizadores*.

Blocking users

Novo na versão 4.7.

In case some users behave badly in your project, you have an option to block them from contributing. The blocked user still will be able to see the project if he has permissions for that, but he won't be able to contribute.

Gestão de permissões por projeto

Pode definir os seus projetos para *Protegido* ou *Privado* e *gerir utilizadores* por projeto na interface de utilizador do Weblate.

Por padrão, isso impede que o Weblate conceda acesso fornecido por *grupos padrão* *Utilizadores* e *Visualizadores* devido à própria configuração desses grupos. Isso não impede que conceda permissões a esses projetos em todo o site, a alterar grupos padrão, a criar um novo ou a criar configurações personalizadas adicionais para componentes individuais, conforme descrito em *Controle de acesso personalizado* abaixo.

Um dos principais benefícios de gerir permissões por meio da interface de utilizador do Weblate é que pode delegá-las a outros utilizadores sem dar a eles o privilégio de superutilizador. Para isso, adicione-os ao grupo *Administração* do projeto.

2.6.2 Controle de acesso personalizado

Nota: Este recurso está indisponível para os projetos que executam o plano Libre no Hosted Weblate.

O sistema de permissão é baseado em grupos e funções, onde as funções de definir um conjunto de permissões e grupos vinculam-nas aos utilizadores e traduções. Veja *Utilizadores*, *papéis*, *grupos* e *permissões* para mais detalhes.

Os recursos mais poderosos do sistema de controle de acesso do Weblate por enquanto estão disponíveis apenas através da *interface administrativa do Django*. Pode usá-la para gerir as permissões de qualquer projeto. Não precisa necessariamente definir o *controle de acesso* para *Personalizado* para utilizá-lo. No entanto, deve ter privilégios de superutilizador para usá-lo.

Se não está interessado em detalhes de implementação e apenas deseja criar uma configuração simples o suficiente com base nos padrões, ou não tem um acesso de todo o site para toda a instalação do Weblate (como em *Hosted Weblate*), consulte a secção *Controle de acesso personalizado*.

Configurações comuns

Esta secção contém uma visão geral de algumas configurações comuns nas quais pode estar interessado.

Gestão de permissões para todo o site

Para gerir permissões para uma instância inteira de uma vez, adicione utilizadores aos *grupos padrão*:

- *Utilizadores* (isso é feito por padrão pela *associação de grupo automática*).
- *Revisores* (se estiver a usar *fluxo de trabalho de revisão* com revisores dedicados).
- *Gestores* (se deseja delegar a maioria das operações de gestão a outra pessoa).

Deve manter todos os projetos configurados como *Públicos* (veja *Controle de acesso ao projeto*), caso contrário as permissões de todo o site fornecidas pela participação nos grupos *Utilizadores* e *Revisores* não terão nenhum efeito.

Também pode conceder algumas permissões adicionais de sua escolha aos grupos padrão. Por exemplo, pode dar permissão para gerir capturas de ecrã para todos os *Utilizadores*.

Também pode definir alguns novos grupos personalizados. Se deseja continuar a gerir as suas permissões em todo o site para esses grupos, escolha um valor apropriado para *Seleção de projeto* (por exemplo *Todos os projetos* ou *Todos os projetos públicos*).

Permissões personalizadas a idiomas, componentes ou projetos

Pode criar os seus próprios grupos dedicados para gerir permissões para objetos distintos, como linguagens, componentes e projetos. Embora esses grupos só possam conceder privilégios adicionais, não pode revogar qualquer permissão concedida por grupos de todo o site ou por projeto a adicionar outro grupo personalizado.

Exemplo:

Se quiser (por qualquer motivo) permitir a tradução para um idioma específico (digamos, *tcheco*) apenas para um conjunto fechado de tradutores confiáveis, a manter as traduções para outros idiomas públicos, terá que:

1. Remover a permissão para traduzir *Tcheco* de todos os utilizadores. Na configuração padrão, isso pode ser feito a alterar o *grupo padrão Utilizadores*.

Table 1: Grupo *Utilizadores*

Seleção de idioma	<i>Como definido</i>
Idiomas	Todos, exceto <i>Tcheco</i>

2. Adicionar um grupo dedicado para tradutores(as) de *Tcheco*.

Table 2: Grupo *Tradutores(as) de tcheco*

Funções	<i>Utilizadores(as) avançados(as)</i>
Seleção de projeto	<i>Todos os projetos públicos</i>
Seleção de idioma	<i>Como definido</i>
Idiomas	<i>Tcheco</i>

3. Adicionar utilizadores aos quais deseja conceder permissões a este grupo.

Como pode ver, a gestão de permissões dessa maneira é poderosa, mas pode ser um trabalho tedioso. Não pode delegá-lo a outro utilizador, a menos que conceda permissões de superutilizador.

Utilizadores, papéis, grupos e permissões

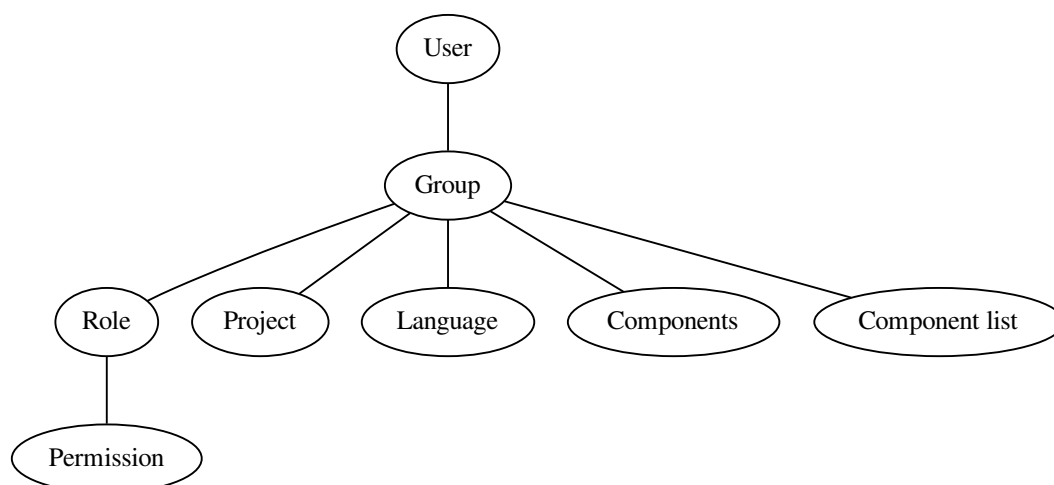
Os modelos de autenticação consistem em vários objetos:

Permissão Permissão individual definida pelo Weblate. Permissões não podem ser atribuídas a utilizadores. Isso só pode ser feito através da atribuição de papéis.

Função Um papel define um conjunto de permissões. Isso permite a reutilização desses conjuntos em vários lugares, a facilitar a administração.

Utilizador Utilizador pode pertencer a vários grupos.

Grupo Grupo conecta papéis, utilizadores e objetos de autenticação (projetos, idiomas e listas de componentes).



Nota: Um grupo não pode ter papéis lhe atribuídos; nesse caso, o acesso para navegar no projeto por qualquer pessoa é presumido (veja abaixo).

Acesso para navegar até um projeto

Um utilizador tem que ser um membro de um grupo vinculado ao projeto ou qualquer componente dentro daquele projeto. Ser membro é suficiente, não são necessárias permissões específicas para navegar o projeto (isso é usado no grupo padrão *Visualizadores*, consulte [Lista de grupos](#)).

Acesso para navegar até um componente

Um utilizador pode acessar componentes irrestritos assim que for possível acessar o projeto dos componentes (e terá todas as permissões que foram concedidas ao utilizador para o projeto). Com o [Acesso restrito](#) ativado, o acesso ao componente requer permissões explícitas para o componente (ou uma lista de componentes na qual o componente está contido).

Escopo de grupos

O escopo da permissão atribuída pelas funções nos grupos é aplicado pelas seguintes regras:

- Se o grupo especificar qualquer *Lista de componentes*, todas as permissões dadas aos membros desse grupo serão concedidas para todos os componentes nas listas de componentes anexadas ao grupo e um acesso sem permissões adicionais é concedido para todos os projetos estes componentes estão contidos. *Componentes* e *Projetos* são ignorados.
- Se o grupo especificar quaisquer *Componentes*, todas as permissões dadas aos membros desse grupo são concedidas para todos os componentes anexados ao grupo e um acesso sem permissões adicionais é concedido para todos os projetos nos quais que esses componentes estão. *Projetos* são ignorados.
- Caso contrário, se o grupo especificar *Projetos*, seja a listar-os diretamente ou a ter *Seleção de projetos* definida com um valor como *Todos os projetos públicos*, todas essas permissões são aplicadas a todos os projetos, que efetivamente concede as mesmas permissões para acessar todos os [componentes irrestritos](#) dos projetos.

- As restrições impostas pelos *Idiomas* de um grupo são aplicadas separadamente, quando é verificado se um utilizador tem acesso para realizar determinadas ações. Ou seja, é aplicado apenas a ações diretamente relacionadas ao próprio processo de tradução, como revisar, gravar traduções, adicionar sugestões, etc.

Dica: Use *Seleção de idioma* ou *Seleção de projeto* para automatizar a inclusão de todos os idiomas ou projetos.

Exemplo:

Digamos que haja um projeto `foo` com os componentes `foo/bar` e `foo/baz` e o seguinte grupo:

Table 3: Grupo *Revisores-admins de espanhol*

Funções	<i>Rever cadeias, Gerir repositório</i>
Componentes	<code>foo/bar</code>
Idiomas	<i>Espanhol</i>

Membros deste grupo terão as seguintes permissões (a presumir as configurações padrão de papéis):

- Acesso geral (navegação) para todo o projeto `foo` incluindo ambos componentes nele: `foo/bar` e `foo/baz`.
- Rever cadeias na tradução para espanhol de `foo/bar` (não em outro lugar).
- Gerir VCS para todo o repositório `foo/bar`, por exemplo, fazer commit de alterações pendentes feitas por tradutores de todos os idiomas.

Atribuições de grupo automáticas

Na parte inferior da página de edição *Grupo* na *interface administrativa do Django*, pode especificar *Associações automáticas de grupo*, que é uma lista de expressões regulares usadas para atribuir automaticamente utilizadores recém-criados a um grupo com base nos endereços de e-mail deles. Esta associação só acontece após a criação da conta.

O caso de uso mais comum para o recurso é atribuir todos os novos utilizadores a algum grupo padrão. Para fazer isso, provavelmente desejará manter o valor padrão (`^.*$`) No campo da expressão regular. Outro caso de uso para essa opção pode ser conceder alguns privilégios adicionais aos funcionários da sua empresa por padrão. Supondo que todos eles usem endereços de e-mail corporativos no seu domínio, isso pode ser feito com uma expressão como `^.*@minhaempresa.com`.

Nota: Associação automática de grupo para *Utilizadores* e *Visualizadores* sempre será recriada ao atualizar de uma versão anterior do Weblate para outra. Se deseja desativá-los, defina a expressão regular para `^$` (que nunca corresponderá).

Nota: Por enquanto, não há como adicionar em massa utilizadores já existentes a algum grupo por meio da interface do utilizador. Para isso, pode recorrer ao uso da *API REST*.

Grupos e funções predefinidos

Após a instalação, um conjunto padrão de grupos é criado (veja [Lista de grupos](#)).

Esses funções e grupos são criados após a instalação. As funções embutidas são sempre mantidas atualizadas pela migração do banco de dados ao atualizar. Não pode alterá-las. Defina uma nova função caso queira definir o seu próprio conjunto de permissões.

Lista de privilégios

Faturamento (consulte [Faturação](#)) Visualizar informações de faturamento [*Administração, Faturamento*]

Alterações Descarragar alterações [*Administração*]

Comentários Publicar comentário [*Administração, Editar fonte, Utilizador avançado, Revisar cadeias, Traduzir*]

Apagar comentário [*Administração*]

Componente Editar configurações do componente [*Administração*]

Bloquear componente, a evitar traduções [*Administração*]

Glossário Adicionar entrada do glossário [*Administração, Gerir glossário, Utilizador avançado*]

Editar entrada do glossário [*Administração, Gerir glossário, Utilizador avançado*]

Apagar entrada do glossário [*Administração, Gerir glossário, Utilizador avançado*]

Enviar entradas do glossário [*Administração, Gerir glossário, Utilizador avançado*]

Sugestões automáticas Usar sugestões automáticas [*Administração, Editar fonte, Utilizador avançado, Rever cadeias, Traduzir*]

Memória de tradução Editar memória de tradução [*Administração, Gerir memória de tradução*]

Apagar memória de tradução [*Administração, Gerir memória de tradução*]

Projetos Editar configurações do projeto [*Administração*]

Gerir acesso do projeto [*Administração*]

Relatórios Descarragar relatórios [*Administração*]

Capturas de ecrã Adicionar captura de ecrã [*Administração, Gerir capturas de ecrã*]

Editar captura de ecrã [*Administração, Gerir capturas de ecrã*]

Apagar captura de ecrã [*Administração, Gerir capturas de ecrã*]

Cadeias fonte Editar informações adicionais da cadeia [*Administração, Editar fonte*]

Cadeias Adicionar nova cadeia [*Administração*]

Remover uma cadeia [*Administração*]

Ignorar verificação com falha [*Administração, Editar fonte, Utilizador avançado, Rever cadeias, Traduzir*]

Editar cadeias [*Administração, Editar fonte, Utilizador avançado, Revisar cadeias, Traduzir*]

Revisar cadeias [*Administração, Revisar cadeias*]

Editar cadeia quando as sugestões são forçadas [*Administração, Rever cadeias*]

Editar cadeias fonte [*Administração, Editar fonte, Utilizador avançado*]

Sugestões Aceitar sugestão [*Administração, Editar fonte, Utilizador avançado, Rever cadeias, Traduzir*]

Adicionar sugestão [*Administração, Editar fonte, Adicionar sugestão, Utilizador avançado, Rever cadeias, Traduzir*]

Apagar sugestão [*Administração, Utilizador avançado*]

Votar em sugestão [*Administração, Editar fonte, Utilizador avançado, Rever cadeias, Traduzir*]

Traduções Adicionar idioma para tradução [*Administração, Utilizador avançado, Gerir idiomas*]

Efetuar tradução automática [*Administração, Gerir idiomas*]

Apagar tradução existente [*Administração, Gerir idiomas*]

Adicionar idiomas para tradução [*Administração, Gerir idiomas*]

Envios Definir autor da tradução enviada [*Administração*]

Sobrescrever cadeias existentes com envio [*Administração, Editar fonte, Utilizador avançado, Rever cadeias, Traduzir*]

Enviar traduções [*Administração, Editar fonte, Utilizador avançado, Rever cadeias, Traduzir*]

VCS Acessar o repositório interno [*Administração, Acessar repositório, Utilizador avançado, Gerir repositório*]

Submeter as alterações ao repositório interno [*Administração, Gerir repositório*]

Fazer push das alterações do repositório interno [*Administração, Gerir repositório*]

Redefinir as alterações no repositório interno [*Administração, Gerir repositório*]

Ver o local do repositório upstream [*Administração, Acessar repositório, Utilizador avançado, Gerir repositório*]

Atualizar o repositório interno [*Administração, Gerir repositório*]

Privilégios para todo o site Utilizar a interface de gestão

Adicionar novos projetos

Adicionar definições de idioma

Gerir definições de idioma

Gerir grupos

Gerir utilizadores

Gerir funções

Gerir anúncios

Gerir a memória de tradução

Gerir as listas de componentes

Nota: Privilégios para todo o site não são concedidos a nenhuma função padrão. Eles são poderosos e muito próximos do estado de superutilizador. A maioria deles afetam todos os projetos da sua instalação do Weblate.

Lista de grupos

Os seguintes grupos são criados após a instalação (ou após a execução de `setupgroups`) e está livre a modificá-los. A migração irá, no entanto, recriá-los se apagá-los ou renomeá-los.

Convidados Define permissões para utilizadores não autenticados.

Este grupo contém apenas utilizadores anônimos (consulte `ANONYMOUS_USER_NAME`).

Pode remover funções deste grupo para limitar as permissões para utilizadores não autenticados.

Funções predefinidas: *Adicionar sugestão, Acessar repositório*

Visualizadores Essa função garante a visibilidade de projetos públicos para todos os utilizadores. Por padrão, todos os utilizadores são membros deste grupo.

Por padrão, *associação automática de grupo* faz com que todos os novos membros de contas deste grupo quando eles entram.

Funções predefinidas: nenhuma

Utilizadores Grupo predefinido para todos os utilizadores.

Por padrão, *associação automática de grupo* faz com que todos os novos membros de contas deste grupo quando eles entram.

Funções predefinidas: *Utilizador avançado*

Revisores Grupo para revisores (consulte *Fluxos de trabalho de tradução*).

Funções predefinidas: *Revisar cadeias*

Gestores Grupo pra administradores.

Funções predefinidas: *Administração*

Aviso: Nunca remova os grupos e utilizadores predefinidos do Weblate, pois isso pode levar a problemas inesperados! Se não tiver uso para eles, pode remover todos os privilégios deles.

2.6.3 Restrições adicionais de acesso

Se quiser usar a instalação do Weblate de uma maneira menos pública, ou seja, permitir novos utilizadores apenas por convite, isso pode ser feito a configurar o Weblate de forma que apenas utilizadores conhecidos tenham acesso a ele. Para fazer isso, pode definir *REGISTRATION_OPEN* para *False* para prevenir registos de quaisquer novos utilizadores e definir *REQUIRE_LOGIN* para */.** para exigir login para acessar todas as páginas do site. Esta é basicamente a forma de bloquear a instalação do Weblate.

Dica: Pode usar os *convites* embutidos para adicionar novos utilizadores.

2.7 Projetos de tradução

2.7.1 Organização de tradução

O Weblate organiza o conteúdo VCS traduzível do projeto/componentes numa estrutura semelhante a uma árvore.

- O objeto de nível inferior é *Configuração de projeto*, que deve conter todas as traduções pertencentes a um conjunto (por exemplo, tradução de uma aplicação em várias versões e/ou documentação que o acompanha).
- No nível acima, *Configuração de componente*, que é na verdade o componente a ser traduzido, define o repositório VCS a ser usado e a máscara dos ficheiros a serem traduzidos.
- Acima da *Configuração de componente* existem traduções individuais, manipuladas automaticamente pelo Weblate como ficheiros de tradução (que correspondem a *Máscara de ficheiros* definido em *Configuração de componente*) aparecem no repositório VCS.

O Weblate tem suporte a uma ampla gama de formatos de tradução (bilíngues e monolíngues) suportados pelo Translate Toolkit, veja *Formatos de ficheiros suportados*.

Nota: Pode compartilhar repositórios VCS clonados a usar *URLs internas do Weblate*. Usar este recurso é altamente recomendado quando tem muitos componentes a compartilhar o mesmo VCS. Ele melhora o desempenho e diminui o espaço em disco necessário.

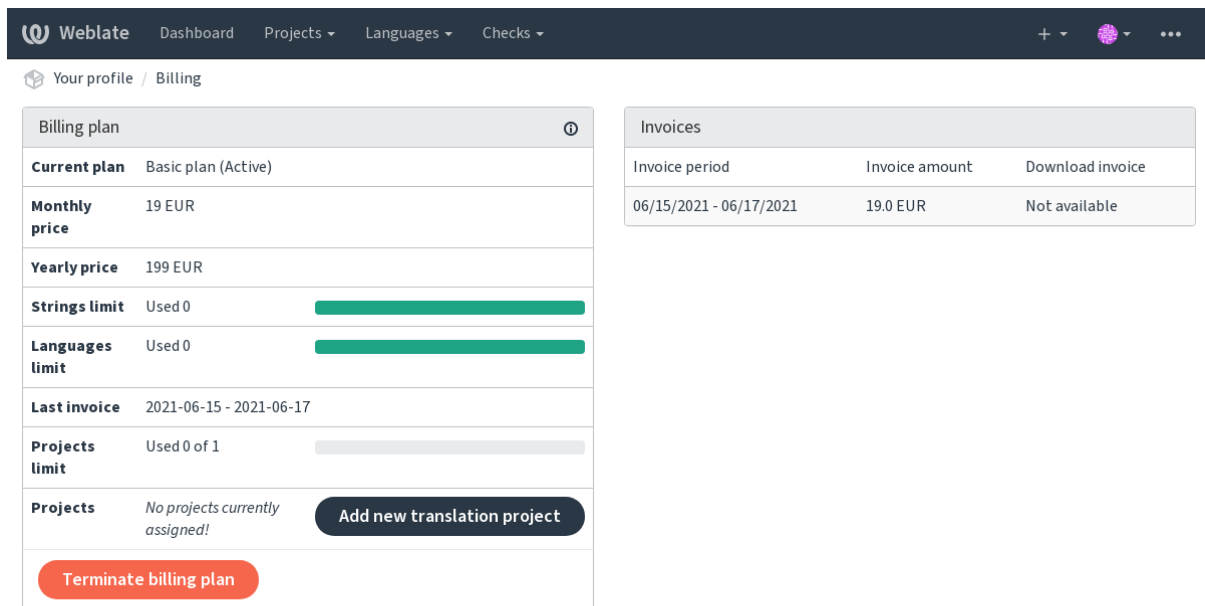
2.7.2 Adicionando projetos e componentes de tradução

Alterado na versão 3.2: Uma interface para adicionar projetos e componentes está incluída e não precisa mais usar *A interface administrativa do Django*.

Alterado na versão 3.4: O processo de adição de componentes agora tem vários estágios, com descoberta automatizada da maioria dos parâmetros.

Com base nas suas permissões, novos projetos e componentes de tradução podem ser criados. É sempre permitido para utilizadores com a permissão *Adicionar novos projetos* e se a sua instância usa cobrança (por exemplo, como <https://hosted.weblate.org/>, veja *Faturação*), também pode criar aqueles com base na permissão dos seus planos da conta de utilizador que gere a cobrança.

Pode visualizar o seu plano de cobrança atual numa página separada:



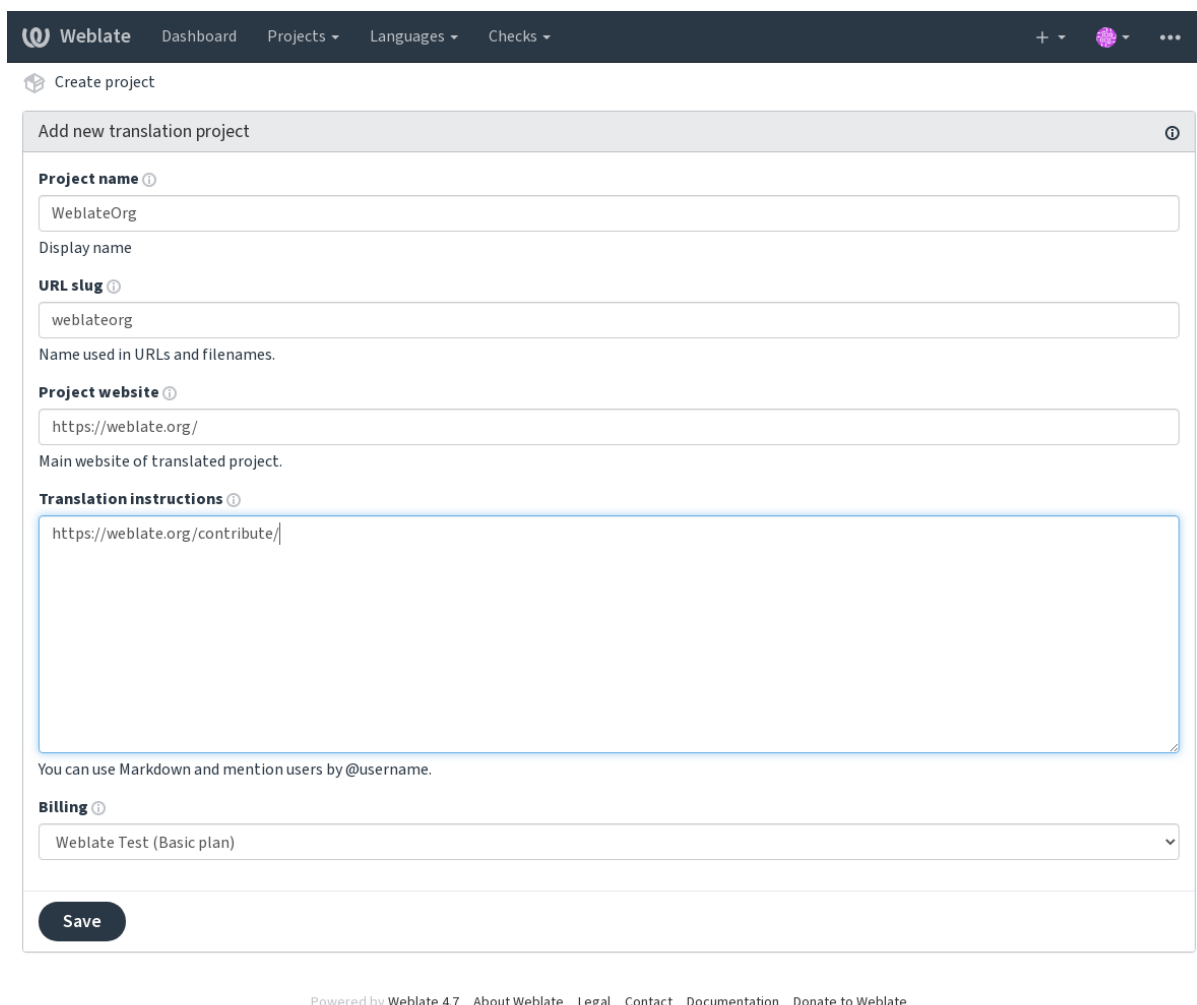
The screenshot shows the Weblate user interface. At the top is a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a breadcrumb 'Your profile / Billing'. The main content is divided into two panels. The left panel, titled 'Billing plan', shows the 'Current plan' as 'Basic plan (Active)'. It lists the 'Monthly price' as 19 EUR and the 'Yearly price' as 199 EUR. It also shows usage for 'Strings limit' and 'Languages limit', both at 0. The 'Last invoice' is for the period 2021-06-15 to 2021-06-17. The 'Projects limit' is 0 of 1. At the bottom of this panel is a red button 'Terminate billing plan'. The right panel, titled 'Invoices', shows the 'Invoice period' as 06/15/2021 - 06/17/2021, the 'Invoice amount' as 19.0 EUR, and the 'Download invoice' link as 'Not available'. At the bottom of the 'Projects' section in the left panel is a dark blue button 'Add new translation project'.

Billing plan	
Current plan	Basic plan (Active)
Monthly price	19 EUR
Yearly price	199 EUR
Strings limit	Used 0
Languages limit	Used 0
Last invoice	2021-06-15 - 2021-06-17
Projects limit	Used 0 of 1
Projects	No projects currently assigned!

Invoices		
Invoice period	Invoice amount	Download invoice
06/15/2021 - 06/17/2021	19.0 EUR	Not available

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

A criação do projeto pode ser iniciada a partir daí, ou a utilizar o menu da barra de navegação, a preencher informações básicas sobre o projeto de tradução para completar a adição dele:

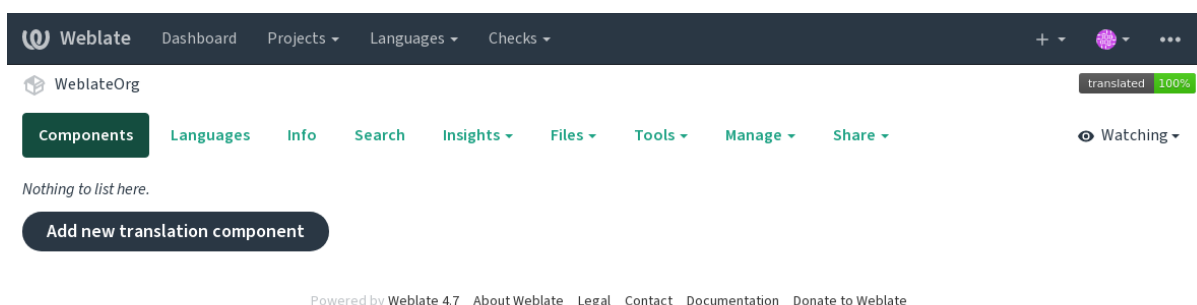


The screenshot shows the 'Add new translation project' form in the Weblate interface. The form is titled 'Add new translation project' and includes several sections:

- Project name:** A text input field containing 'WeblateOrg'.
- Display name:** A text input field.
- URL slug:** A text input field containing 'weblateorg'.
- Project website:** A text input field containing 'https://weblate.org/'.
- Translation instructions:** A large text area containing 'https://weblate.org/contribute/'.
- Billing:** A dropdown menu showing 'Weblate Test (Basic plan)'.

At the bottom of the form is a 'Save' button. Below the form, there is a footer with the text 'Powered by Weblate 4.7' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Depois de criar o projeto, é levado diretamente à página do projeto:



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 section for 'WeblateOrg' with a 'translated 100%' indicator. The main content area shows a 'Components' tab selected, with a list of components. Below the list, there is a button 'Add new translation component'. The footer of the page includes the text 'Powered by Weblate 4.7' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

A criação de um novo componente de tradução pode ser iniciada com um único clique lá. O processo de criação de um componente possui vários estágios e detecta automaticamente a maioria dos parâmetros de tradução. Existem várias abordagens para a criação de componentes:

Do controle de versão Cria um componente do repositório de controle de versão remoto.

Do componente existente Cria um componente adicional ao existente, a escolher ficheiros diferentes.

Ramo adicional Cria um componente adicional ao existente, apenas para um ramo diferente.

Enviar ficheiros de tradução Envia os ficheiros de tradução para o Weblate caso não tenha controle de versão ou não queira integrá-lo ao Weblate. Pode atualizar o conteúdo posteriormente a usar a interface web ou [API REST do Weblate](#).

Traduzir documento Envia um único documento e o traduz.

Começar do zero Cria um projeto de tradução vazio e adiciona cadeias manualmente.

Depois de ter os componentes de tradução existentes, também pode adicionar facilmente novos para ficheiros adicionais ou ramos a usar o mesmo repositório.

Primeiro precisa preencher o nome e a localização do repositório:

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 with translations.

Source code repository ⓘ

https://github.com/WeblateOrg/demo.git

URL of a repository, use weblate://project/component for sharing with other component.

Repository branch ⓘ

Repository branch to translate

Continue

Powered by Weblate 4.7 About Weblate Legal Contact Documentation Donate to Weblate

Na próxima página, verá uma lista de recursos traduzíveis descobertos:

Create component

Add new translation component ⓘ

Choose translation files to import ⓘ

☐ Specify configuration manually

☐ File format Android String Resource, Filemask app/src/main/res/values-*/strings.xml

☐ File format gettext PO file, Filemask weblate/langdata/locale/*/LC_MESSAGES/django.po


☐ File format gettext PO file, Filemask weblate/locale/*/LC_MESSAGES/django.po

☐ File format gettext PO file, Filemask weblate/locale/*/LC_MESSAGES/djangojs.po

Continue

Powered by Weblate 4.7 About Weblate Legal Contact Documentation Donate to Weblate

Como última etapa, analisa as informações do componente de tradução e preenche os detalhes opcionais:

 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.

File format

gettext PO file

Filemask

weblate/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

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 filemask.

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 4.7
 [About Weblate](#)
[Legal](#)
[Contact](#)
[Documentation](#)
[Donate to Weblate](#)

Veja também:

A interface administrativa do Django, Configuração de projeto, Configuração de componente

2.7.3 Configuração de projeto

Cria um projeto de tradução e, em seguida, adiciona um novo componente para tradução nele. O projeto é como uma estante, na qual as traduções reais estão empilhadas. Todos os componentes no mesmo projeto compartilham sugestões e o dicionário deles; as traduções também são propagadas automaticamente por todos os componentes num único projeto (a menos que desligado na configuração do componente), veja *Memória de Tradução*.

Veja também:

/devel/integration

Esses atributos básicos configuram e informam os tradutores de um projeto:

Nome do projeto

Nome detalhado do projeto, usado para exibir o nome do projeto.

URL amigável

Nome do projeto adequado para URLs.

Site da Web do Projeto

URL onde os tradutores podem encontrar mais informações sobre o projeto.

Este é um parâmetro obrigatório, a menos que seja desligado por *WEBSITE_REQUIRED*.

Instruções para tradução

URL para mais sites com instruções mais detalhadas para tradutores.

Definir cabeçalho «Language-Team»

Se o Weblate deve gerir o cabeçalho Language-Team (este é um recurso apenas do *GNU gettext* no momento).

Utilizar memória de tradução partilhada

Para usar memória de tradução compartilhada, veja *Memória de tradução compartilhada* para mais detalhes.

O valor padrão é determinado por *DEFAULT_SHARED_TM*.

Contribuir à memória de tradução compartilhada

Para contribuir com a memória de tradução compartilhada, veja *Memória de tradução compartilhada* para mais detalhes.

O valor padrão é determinado por `DEFAULT_SHARED_TM`.

Controlo de acesso

Configura o controle de acesso por projeto, veja *Controlo de acesso ao projeto* para mais detalhes.

O valor padrão pode ser alterado por `DEFAULT_ACCESS_CONTROL`.

Activar revisões

Ativa o fluxo de trabalho de revisão para traduções, veja *Revisores dedicados*.

Ativar revisões de fontes

Ativa o fluxo de trabalho de revisão para cadeia fonte, veja *Revisões de cadeias fonte*.

Veja também:

report-source, *Comentários*

Ativar hooks

Se *Hooks de notificação* não autenticados devem ser usados para este repositório.

Veja também:

Ficheiro de idioma intermédio, *Portal de qualidade para cadeias fonte*, *Formatos bilíngues e monolíngues*, *Definições de idioma*

Aliases do idioma

Define o mapeamento dos códigos de idioma ao importar traduções para o Weblate. Use isso quando os códigos de idioma forem inconsistentes nos seus repositórios e quiser obter uma visualização consistente no Weblate ou no caso de desejar usar uma nomenclatura fora do padrão dos seus ficheiros de tradução.

O caso de uso típico pode ser mapear inglês americano para inglês: `en_US:en`

Vários mapeamentos devem ser separados por vírgula: `en_GB:en,en_US:en`

Usando código não padrão: `ia_FOO:ia`

Dica: Os códigos de idioma são mapeados ao combinar os ficheiros de tradução e as correspondências diferenciam maiúsculas de minúsculas, por isso certifique-se de usar os códigos de idioma fonte na mesma forma usada nos nomes dos ficheiros.

Veja também:

Analisando códigos de idioma

2.7.4 Configuração de componente

Um componente é um agrupamento de algo para tradução. Insere um local de repositório VCS e uma máscara de ficheiros para os ficheiros que deseja traduzir e o Weblate busca automaticamente neste VCS e encontra todos os ficheiros traduzíveis correspondentes.

Veja também:

`/devel/integration`

Pode encontrar alguns exemplos de configurações típicas nos *Formatos de ficheiros suportados*.

Nota: Recomenda-se manter os componentes de tradução num tamanho razoável – divida a tradução por qualquer coisa que faça sentido no seu caso (apps ou complementos individuais, capítulos de livros ou sites).

O Weblate lida facilmente com traduções com 10000s de cadeias, mas é mais difícil dividir o trabalho e coordenar entre os tradutores com componentes de tradução tão grandes.

Se a definição do idioma de uma tradução estiver a faltar, uma definição vazia é criada e nomeada como «cs_CZ (generated)». Deve ajustar a definição e relatar isso aos autores do Weblate, para que os idiomas em falta possam ser incluídos na próxima versão.

O componente contém todos os parâmetros importantes para trabalhar com o VCS e para obter traduções dele:

Nome do componente

Nome detalhado do componente, usado para exibir o nome do componente.

URL amigável do componente

Nome do componente adequado para URLs.

Projeto do componente

Configuração de projeto ao qual o componente pertence.

Sistema de controlo de versões

VCS para usar, veja *Integração de controlo de versões* para detalhes.

Veja também:

Fazendo push das alterações do Weblate

Repositório do código-fonte

Repositório VCS usado para fazer pull das alterações.

Veja também:

Veja *Acessando repositórios* para mais detalhes sobre especificar URLs.

Dica: Pode ser uma URL VCS real ou `weblate://projeto/componente` a indicar que o repositório deve ser compartilhado com outro componente. Veja *URLs internas do Weblate* para mais detalhes.

URL de submissão do repositório

URL do repositório usado para fazer push. Esta configuração é usada apenas para *Git* e *Mercurial* e o suporte a push é desativado para estes quando está vazio.

Veja também:

Veja *Acessando repositórios* para mais detalhes sobre como especificar uma URL de repositório e *Fazendo push das alterações do Weblate* para mais detalhes sobre como fazer push de mudanças do Weblate.

Navegador do repositório

URL de navegação do repositório usada para exibir os ficheiros fonte (localização das mensagens usadas). Quando vazio, nenhuma dessas ligações será gerado. Pode usar *Template markup*.

Por exemplo, no GitHub, use algo como: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

No caso dos seus caminhos serem relativos a pastas diferentes, pode querer remover o diretório principal pelo filtro `parentdir` (veja *Template markup*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

URL do repositório exportado

URL para onde as alterações feitas pelo Weblate são exportadas. Isso é importante quando *Tradução contínua* não é usada, ou quando há a necessidade de mesclar manualmente as alterações. Pode usar *Exportador git* para automatizar isso para repositórios Git.

Ramo do repositório

De qual ramo fazer checkout do VCS e onde procurar traduções.

Ramo do push

Ramo para fazer push de alterações, deixe vazio para usar *Ramo do repositório*.

Nota: Atualmente, isso é compatível apenas com Git, GitLab e GitHub, mas é ignorado para outras integrações VCS.

Veja também:

Fazendo push das alterações do Weblate

Máscara de ficheiros

Máscara de ficheiros a traduzir, incluindo caminho. Deve incluir um «*» a substituir o código de idioma (veja *Definições de idioma* para informações sobre como isso é processado). Caso seu repositório contenha mais de um ficheiro de tradução (por exemplo, mais domínios gettext), precisa criar um componente para cada um deles.

Por exemplo, `po/*.po` ou `locale/*/LC_MESSAGES/django.po`.

No caso do seu nome de ficheiro conter caracteres especiais como `[`, `]`, eles precisam ser escapados como `[]` ou `[]`.

Veja também:

Formatos bilíngues e monolíngues, O que significa «Existem mais ficheiros para o idioma único (en)»?

Ficheiro de idioma base monolingue

Ficheiro base a conter definições de cadeias para *Componentes monolínguas*.

Veja também:

Formatos bilíngues e monolíngues, O que significa «Existem mais ficheiros para o idioma único (en)»?

Editar ficheiro base

Permite a edição do ficheiro base para *Componentes monolínguas*.

Ficheiro de idioma intermédio

Ficheiro de idioma intermediário para *Componentes monolínguas*. Na maioria dos casos, este é um ficheiro de tradução fornecido por programadores e é usado ao criar cadeias de origem reais.

Quando definido, as cadeias fonte são baseados neste ficheiro, mas todos os outros idiomas são baseados em *Ficheiro de idioma base monolingue*. Caso a cadeia não seja traduzida para o idioma fonte, a tradução para outros idiomas é proibida. Isso fornece uma *Portal de qualidade para cadeias fonte*.

Veja também:

Portal de qualidade para cadeias fonte, Formatos bilíngues e monolíngues, O que significa «Existem mais ficheiros para o idioma único (en)»?

Modelo para novas traduções

Ficheiro base usado para gerar novas traduções, por exemplo, um ficheiro `.pot` com `gettext`.

Dica: Em muitos formatos monolíngues, o Weblate começa com um ficheiro em branco por padrão. Use-o no caso de desejar que todas as cadeias estejam presentes com um valor vazio ao criar uma tradução.

Veja também:

adding-translation, Adicionando novas traduções, Adicionar nova tradução, Formatos bilíngues e monolíngues, O que significa «Existem mais ficheiros para o idioma único (en)»?

Formato de ficheiro

Formato do ficheiro de tradução, veja também *Formatos de ficheiros suportados*.

Endereço para reportar erros na cadeia fonte

Endereço de e-mail usado para relatar bugs upstream. Este endereço também receberá notificação sobre quaisquer comentários da cadeia fonte feitos no Weblate.

Permitir propagação da tradução

Pode desativar a propagação de traduções para este componente de outros componentes dentro do mesmo projeto. Isso realmente depende do que está a traduzir, às vezes é desejável ter feito uso de uma tradução mais de uma vez.

Normalmente, é uma boa ideia desligar isso para traduções monolíngues, a menos que esteja a usar os mesmos IDs em todo o projeto.

O valor padrão pode ser alterado por `DEFAULT_TRANSLATION_PROPAGATION`.

Ativar sugestões

Se sugestões de tradução são aceites para este componente.

Votação de sugestão

Ativa a emissão de votos para sugestões, veja *Votação de sugestão*.

Aceitar sugestões automaticamente

Aceita automaticamente sugestões votadas, veja *Votação de sugestão*.

Marcadores de tradução

Personalização de verificações de qualidade e outros comportamentos do Weblate, veja *Personalizando comportamento a usar marcadores*.

Verificações impostas

Lista de verificações que não podem ser ignoradas, veja *Forçar verificações*.

Nota: Forçar a verificação não a ativa automaticamente, ainda deve ativá-la *Personalizando comportamento a usar marcadores* em *Marcadores de tradução* ou *Informações adicionais sobre cadeias fonte*.

Licença da tradução

Licença da tradução (não precisa ser igual à licença do código-fonte).

Acordo de contribuidor

Acordo do utilizador que tem de ser aprovado antes do utilizador poder traduzir este componente.

Adicionar nova tradução

Como lidar com solicitações de criação de novos idiomas. Opções disponíveis:

Contactar gestores O utilizador pode seleccionar o idioma desejado e os mantenedores do projeto receberão uma notificação sobre isso. Cabe adiciona-los (ou não) o idioma ao repositório.

Apontar para URL com instruções de tradução É apresentado ao utilizador uma ligação para a página que descreve o processo de iniciar novas traduções. Use-o caso deseje um processo mais formal (por exemplo, formar uma equipa antes de iniciar a tradução real).

Criar novo ficheiro de idioma O utilizador pode seleccionar o idioma e o Weblate cria automaticamente o ficheiro para ele e a tradução pode começar.

Desativar adição de novas traduções Não haverá a opção para o utilizador iniciar uma nova tradução.

Dica: The project admins can add new translations even if it is disabled here when it is possible (either [Modelo para novas traduções](#) or the file format supports starting from an empty file).

Veja também:

adding-translation, [Adicionando novas traduções](#)

Gerir cadeias

Novo na versão 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.

Para formatos monolíngues, as cadeias são geridas apenas no idioma fonte e são automaticamente adicionadas ou removidas nas traduções. As cadeias aparecem nos ficheiros de tradução assim que são traduzidas.

Veja também:

[Formatos bilíngues e monolíngues](#), [adding-new-strings](#), `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Estilo de código de idioma

Personalizar o código de idioma usado para gerar o nome do ficheiro para traduções criadas por Weblate.

Veja também:

[Adicionando novas traduções](#), [Código do idioma](#), [Analisando códigos de idioma](#)

Estilo de união

You can configure how updates from the upstream repository are handled. This might not be supported for some VCSs. See *Mesclar ou rebase* for more details.

Default value can be changed by `DEFAULT_MERGE_STYLE`.

Mensagens de commit, add, merge e extensão

Message used when committing a translation, see *Template markup*.

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

Enviar ao submeter

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 *Commits adiados*). To actually enable pushing *Repository push URL* has to be configured as well.

Idade das alterações a fazer commit

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`.

Dica: There are other situations where pending changes might be committed, see *Commits adiados*.

Bloquear com erro

Locks the component (and linked components, see *URLs internas do 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 fonte

Language used for source strings. Change this if you are translating from something else than English.

Dica: 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 *Ficheiro de idioma intermédio*.

Filtro de idioma

Expressão regular usada para filtrar a tradução ao examinar por uma máscara de ficheiro. Pode ser usada para limitar a lista de idiomas geridos pelo Weblate.

Nota: You need to list language codes as they appear in the filename.

Some examples of filtering:

Filter description	Expressão regular
Selected languages only	<code>^(cs de es)\$</code>
Exclude languages	<code>^(?! (it fr)\$) .+\$</code>
Filter two letter codes only	<code>^[.].+\$</code>
Exclude non language files	<code>^(?! (blank)\$) .+\$</code>
Include all files (default)	<code>^[^.] +\$</code>

Expressão regular das 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 managers, in the Weblate interface.

Veja também:

O Weblate tem suporte a outros VCSs que não o Git e o Mercurial?, alerts

Prioridade

Componentes de prioridade mais elevada são oferecidos primeiro aos tradutores.

Acesso restrito

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`.

Dica: This applies to project admins as well — please make sure you will not loose access to the component after toggling the status.

Compartilhar em projetos

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.

Usar como um glossário

Novo na versão 4.5.

Permite usar este componente como glossário. Pode configurar como ele será listado a usar a *Cor do glossário*.

The glossary will be accessible in all projects defined by *Compartilhar em projetos*.

It is recommended to enable *Gerir cadeias* on glossaries in order to allow adding new words to them.

Veja também:

Glossário

Cor do glossário

Display color for a glossary used when showing word matches.

2.7.5 Template markup

Weblate uses simple markup language in several places where text rendering is needed. It is based on *The Django template language*, so it can be quite powerful.

Currently it is used in:

- Commit message formatting, see *Configuração de componente*
- **Several addons**
 - *Descoberta de componentes*
 - *Gerador de estatísticas*
 - *Escrevendo scripts para extensões*

There following variables are available in the component templates:

```
{{ language_code }} Código do idioma
{{ language_name }} Nome do idioma
{{ component_name }} Nome do componente
{{ component_slug }} URL amigável do componente
{{ project_name }} Nome do projeto
{{ project_slug }} URL amigável do projeto
{{ url }} Translation URL
{{ filename }} Nome do ficheiro de tradução
{{ stats }} Translation stats, this has further attributes, examples below.
{{ stats.all }} Total strings count
{{ stats.fuzzy }} Count of strings needing review
```

`{{ stats.fuzzy_percent }}` Percent of strings needing review
`{{ stats.translated }}` Translated strings count
`{{ stats.translated_percent }}` Translated strings percent
`{{ stats.allchecks }}` Number of strings with failing checks
`{{ stats.allchecks_percent }}` Percent of strings with failing checks
`{{ author }}` Author of current commit, available only in the commit scope.
`{{ addon_name }}` Name of currently executed addon, available only in the addon commit message.

The following variables are available in the repository browser or editor templates:

`{{branch}}` current branch
`{{line}}` line in file
`{{filename}}` filename, you can also strip leading parts using the `parentdir` filter, for example `{{filename|parentdir}}`

You can combine them with filters:

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

You can use conditions:

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

There is additional tag available for replacing characters:

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

You can combine it with filters:

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

There are also additional filter to manipulate with filenames:

```
Directory of a file: {{ filename|dirname }}  
File without extension: {{ filename|striptext }}  
File in parent dir: {{ filename|parentdir }}  
It can be used multiple times: {{ filename|parentdir|parentdir }}
```

...and other Django template features.

2.7.6 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 *Configuração de produção* for more details, especially:

- Configure Celery for executing background tasks (see *Tarefas de fundo a usar o Celery*)
- *Ativar o cache*
- *Usar um poderoso mecanismo de banco de dados*
- *Desativar o modo de depuração*

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 *Tarefas de fundo a usar o Celery*).

Disable unneeded checks

Some quality checks can be quite expensive, and if not needed, can save you some time during import if omitted. See `CHECK_LIST` for info on configuration.

2.7.7 Automatic creation of components

In case your project has dozen of translation files (e.g. for different gettext domains, or parts of Android apps), you might want to import them automatically. This can either be achieved from the command line by using `import_project` or `import_json`, or by installing the *Descoberta de componentes* addon.

To use the addon, you first need to create a component for one translation file (choose the one that is the least likely to be renamed or removed in future), and install the addon on this component.

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

Veja também:

Comandos de gerência, Descoberta de componentes

2.8 Definições de idioma

Para apresentar diferentes traduções corretamente, são necessárias informações sobre nome do idioma, direção de texto, definições plurais e código de idioma.

2.8.1 Analisando códigos de idioma

Ao analisar as traduções, o Weblate tenta mapear o código de idioma (geralmente o ISO 639-1) para qualquer objeto de idioma existente.

Pode ajustar ainda mais esse mapeamento no nível do projeto por *Aliases do idioma*.

Se nenhuma correspondência exata for encontrada, uma tentativa será feita para melhor encaixá-la num idioma existente. São testadas etapas a seguir:

- Procuras sem diferenciar maiúsculo de minúsculo.
- Normalizar sublinhados e traços.
- Procurar aliases de idioma embutido.
- Procurar por nome de idioma.
- Ignorar o código de país padrão para o idioma dado – escolher `cs` em vez de `cs_CZ`.

Se isso também falhar, uma nova definição de idioma será criada a usar seu padrão (direção do texto da esquerda para a direita, um plural). A linguagem criada automaticamente com o código `xx_XX` será nomeada como `xx_XX (generated)`. Podia mudar isso na interface administrativa mais tarde, (veja [Alterando as definições de idioma](#)) e relatar ao rastreador de problemas (veja [Contribuir ao Weblate](#)), para que a definição apropriada possa ser adicionada ao próximo lançamento do Weblate.

Dica: No caso de ver algo indesejado como idioma, podia ajustar [Filtro de idioma](#) para ignorar tal ficheiro ao analisar traduções.

Veja também:

[Código do idioma](#), [Adicionando novas traduções](#)

2.8.2 Alterando as definições de idioma

Pode alterar as definições de idioma na interface de idiomas (URL `/languages/`).

Durante a edição, certifique-se de que todos os campos estejam corretos (especialmente os plurais e a direção do texto), caso contrário, os tradutores não poderão editar adequadamente essas traduções.

2.8.3 Definições embutidas de idioma

Definições para cerca de 600 idiomas estão incluídas no Weblate e a lista é estendida a cada versão. Sempre que o Weblate é atualizado (mais especificamente sempre que **weblate migrate** é executado, veja [Instruções genéricas de atualização](#)) o banco de dados de idiomas é atualizado para incluir todas as definições de idioma enviadas no Weblate.

Este recurso pode ser desativado a usar `UPDATE_LANGUAGES`. Também pode forçar a atualização do banco de dados para corresponder aos dados embutidos do Weblate a usar `setuplang`.

Veja também:

As definições de idioma estão no repositório [weblate-language-data repository](#).

2.8.4 Códigos de idioma ambíguos e macrolinguagens

Em muitos casos, não é uma boa ideia usar código de idioma macro para uma tradução. O caso típico problemático pode ser a língua curda, que pode ser escrita em árabe ou latim, a depender da variante real. Para obter o comportamento correto no Weblate, é recomendável usar apenas códigos de idioma individuais e evitar macroidiomas.

Veja também:

[Definição de macroidiomas](#), [Lista de macroidiomas](#)

2.8.5 Definições de idioma

Cada idioma consiste nos seguintes campos:

Código do 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](#).

Veja também:

Analisando códigos de idioma, Adicionando novas traduções

Nome do idioma

Nome visível do idioma. Os nomes dos idiomas incluídos no Weblate também estão a ser localizados a depender do idioma da interface do utilizador.

Direção do texto

Determina se o idioma é escrito da direita para a esquerda ou da esquerda para a direita. Esta propriedade é autode-tectada corretamente para a maioria dos idiomas.

Quantidade de plurais

Quantidade de plurais usados no idioma.

Fórmula de plural

Fórmula de plural compatível com Gettext usada para determinar qual forma de plural é usada para determinada contagem.

Veja também:

Plurais, [GNU gettext utilities: Plural forms](#), [Language Plural Rules by the Unicode Consortium](#)

2.8.6 Adicionando novas traduções

Alterado na versão 2.18: Nas versões anteriores a 2.18, o comportamento de adicionar novas traduções era específico do formato de ficheiro.

Weblate pode iniciar automaticamente uma nova tradução para todos os formatos de ficheiro.

Alguns formatos esperam começar com um ficheiro vazio e apenas cadeias traduzidas a serem incluídas (por exemplo, *Recurso de cadeias de Android*), enquanto outros esperam ter todas as chaves presentes (por exemplo, *GNU gettext*). Em algumas situações, isso realmente não depende do formato, mas sim do framework que usa para lidar com a tradução (por exemplo, com *Ficheiro JSON*).

Quando especifica *Modelo para novas traduções* em *Configuração de componente*, Weblate usará este ficheiro para iniciar novas traduções. Todas as traduções existentes serão removidas do ficheiro ao fazer isso.

Quando *Modelo para novas traduções* está vazio e o formato de ficheiro suporta, um ficheiro vazio é criado onde novas cadeias serão adicionadas assim que forem traduzidos.

O *Estilo de código de idioma* permite que personalize o código de idioma usado nos nomes de ficheiros gerados:

Predefinição baseada no formato do ficheiro Dependendo do formato do ficheiro, para a maioria deles utiliza-se POSIX.

Estilo POSIX utilizando o sublinhado como um separador Normalmente usado por gettext e ferramentas relacionadas, produz códigos de linguagem como `pt_BR`.

Estilo de POSIX utilizando o sublinhado como um separador, incluindo o código do país Código de idioma do estilo POSIX incluindo o código do país mesmo quando não necessário (por exemplo, `cs_CZ`).

Estilo BCP utilizando o hífen como um separador Normalmente usado em plataformas web, produz códigos de idioma como `pt-BR`.

Estilo de BCP utilizando o hífen como um separador, incluindo o código do país Código de idioma do estilo BCP, incluindo o código do país, mesmo quando não necessário (por exemplo, `cs-CZ`).

Estilo Android Usado apenas em apps Android, produz códigos de idioma como `pt-rBR`.

Estilo Java Usado por Java – principalmente BCP com códigos legados para chinês.

Além disso, quaisquer mapeamentos definidos em *Aliases do idioma* são aplicados ao contrário.

Nota: O Weblate reconhece qualquer um desses ao analisar ficheiros de tradução; as configurações acima apenas influenciam como os novos ficheiros são criados.

Veja também:

Código do idioma, Analisando códigos de idioma

2.9 Tradução contínua

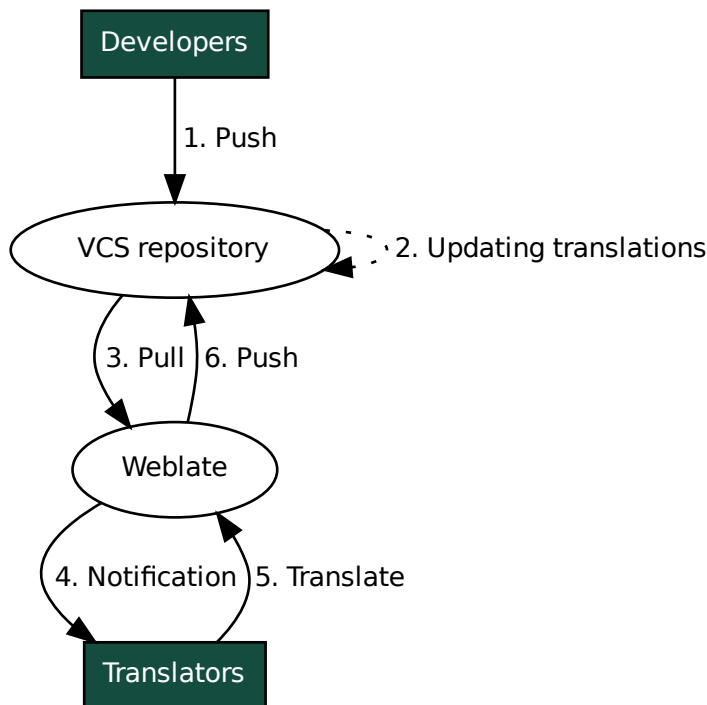
Há infraestrutura em vigor para que a sua tradução acompanhe o desenvolvimento de perto . Desta forma, os tradutores podem trabalhar em traduções o tempo todo, em vez de trabalhar com uma enorme quantidade de texto novo pouco antes do lançamento.

Veja também:

`/devell/integration` descreve maneiras básicas de integrar o seu desenvolvimento com o Weblate.

O processo é o seguinte:

1. Os programadores fazem alterações e fazem *push* delas so repositório VCS.
2. Opcionalmente, os ficheiros de tradução são atualizados (isso depende do formato do ficheiro, consulte *Por que é que o Weblate ainda mostra as cadeias de tradução antigas quando atualizei o modelo?*).
3. O Weblate faz o *pull* das alterações do repositório VCS, consulte *Atualizar repositórios*.
4. Uma vez que o Weblate deteta alterações nas traduções, os tradutores são notificados com base na configurações de assinatura deles.
5. Os tradutores enviam traduções a usar a interface web do Weblate ou enviam alterações feitas offline.
6. Uma vez que os tradutores acabaram, o Weblate faz commit das alterações no repositório local (veja *Commits adiados*) e faz *push* delas de volta se tiver permissões para fazê-lo (veja *Fazendo push das alterações do Weblate*).



2.9.1 Atualizar repositórios

Deve configurar alguma maneira de atualizar repositórios de backend a partir da fonte dele.

- Use *Hooks de notificação* para integrar com a maioria dos serviços comuns de hospedagem de código:
 - *Receber alterações do GitHub automaticamente*
 - *Receber alterações do GitLab automaticamente*
 - *Receber alterações do Bitbucket automaticamente*
 - *Receber alterações do Pagure automaticamente*
 - *Receber alterações dos Azure Repos automaticamente*
- Acione manualmente a atualização na gestão do repositório ou a usar *API REST do Weblate* ou *Cliente Weblate*
- Ative `AUTO_UPDATE` para atualizar todos os componentes na sua instância Weblate automaticamente
- Execute `updategit` (com a seleção de um projeto ou `--all` para atualizar tudo)

Sempre que o Weblate atualiza o repositório, as extensões de pós-atualização serão acionadas, consulte *Extensões*.

Evitar conflitos de mesclagem

Os conflitos de mesclagem do Weblate surgem quando o mesmo ficheiro foi alterado tanto no Weblate quanto fora dele. Existem duas abordagens para lidar com isso - evitar edições fora do Weblate ou integrar o Weblate no seu processo de atualização, de modo que descarte alterações antes de atualizar os ficheiros fora do Weblate.

A primeira abordagem é fácil com ficheiros monolíngues - pode adicionar novas cadeias no Weblate e deixar a edição completa dos ficheiros lá. Para ficheiros bilíngues, geralmente há algum tipo de processo de extração de mensagens para gerar ficheiros traduzíveis do código-fonte. Em alguns casos, isso pode ser dividido em duas partes - uma para a extração gera modelo (por exemplo, o GETTEXT POT é gerado a usar **xgettext**) e depois o processo a seguir mescla-o em traduções reais (os ficheiros GETTEXT PO são atualizados a usar **msgmerge**). Pode executar o segundo passo dentro do Weblate e garantirá que todas as alterações pendentes sejam incluídas antes desta operação.

A segunda abordagem pode ser alcançada a usar o *API REST do Weblate* para forçar o Weblate a fazer push de todas as alterações pendentes e bloquear a tradução enquanto está fazendo alterações do seu lado.

O script para fazer atualizações pode ser assim:

```
# 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
```

Se tiver vários componentes compartilhando o mesmo repositório, deve bloqueá-los todos separadamente:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Nota: O exemplo usa *Cliente Weblate*, que precisa de configuração (chaves de API) para ser capaz de controlar o Weblate remotamente. Também pode conseguir isso a usar qualquer cliente HTTP em vez de wlc, por exemplo, curl, ver *API REST do Weblate*.

Veja também:

Cliente Weblate

Receber alterações do GitHub automaticamente

O Weblate vem com suporte nativo ao GitHub.

Se estiver a usar o Hosted Weblate, a abordagem recomendada é instalar o app *Weblate*, dessa forma terá a configuração correta sem ter que configurar muito. Também pode ser usado para fazer push de mudanças de volta.

Para receber notificações em cada push a um repositório do GitHub, adicione o webhook do Weblate nas configurações do repositório (*Webhooks*) como mostrado na imagem abaixo:

The screenshot shows the GitHub interface for the repository 'WeblateOrg / hello'. The 'Settings' tab is selected, and the 'Webhooks' section is active. The 'Add webhook' form is displayed with the following details:

- Payload URL:** `https://hosted.weblate.org/hooks/github/`
- Content type:** `application/x-www-form-urlencoded`
- Secret:** (Empty text field)
- SSL verification:** A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' is checked. A red button 'Disable SSL verification' is visible.
- Which events would you like to trigger this webhook?:**
 - ☒ Just the push event.
 - ☐ Send me everything.
 - ☐ Let me select individual events.
- Active:** A checkbox is checked, with the text 'We will deliver event details when this hook is triggered.'
- Add webhook:** A green button at the bottom of the form.

The footer of the page includes copyright information for GitHub, Inc. (© 2018), links to Terms, Privacy, Security, Status, and Help, and a GitHub logo. On the right, there are links to Contact GitHub, API, Training, Shop, Blog, and About.

Para a URL de carga útil, anexar `/hooks/github/` à URL do Weblate, por exemplo, para o serviço Hosted Weblate, este é `https://hosted.weblate.org/hooks/github/`.

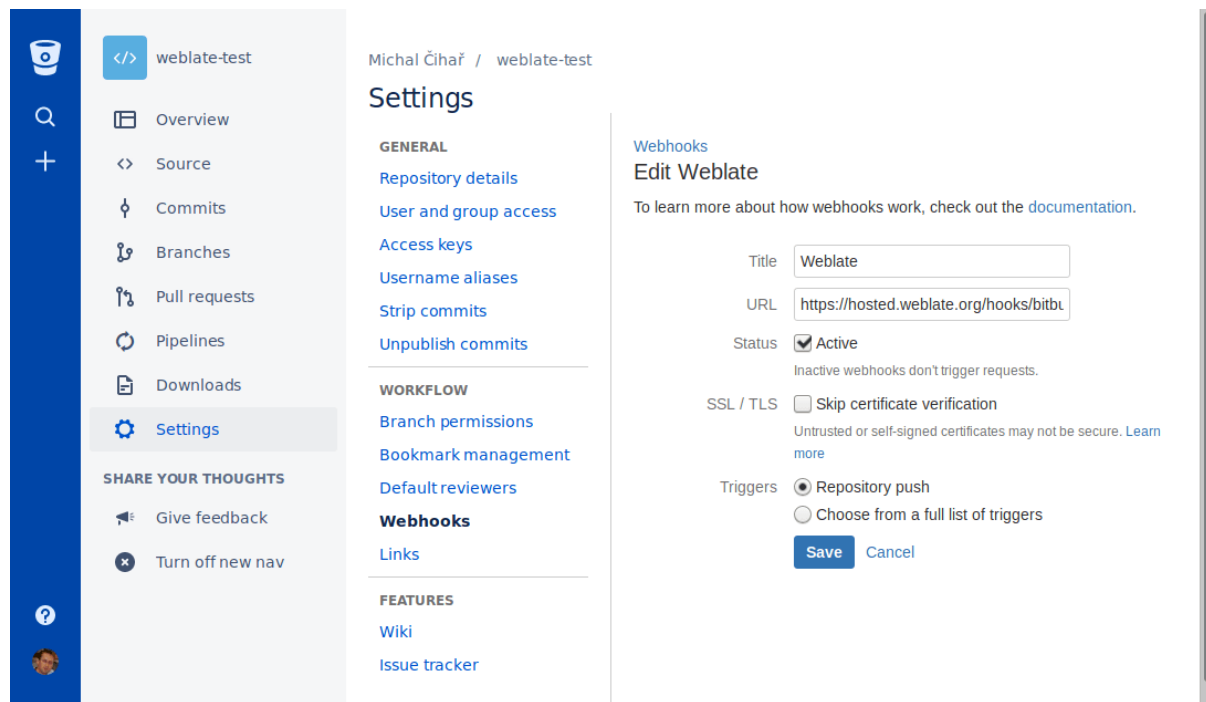
Pode deixar outros valores nas configurações predefinidas (o Weblate pode lidar com ambos os tipos de conteúdo e consome apenas o evento *push*).

Veja também:

POST `/hooks/github/`, Acessando repositórios do Hosted Weblate

Receber alterações do Bitbucket automaticamente

O Weblate tem suporte para webhooks do Bitbucket, adicione um webhook que aciona no push do repositório, com destino a URL `/hooks/bitbucket/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/bitbucket/`).

**Veja também:**

POST /hooks/bitbucket/, Acessando repositórios do Hosted Weblate

Receber alterações do GitLab automaticamente

O Weblate tem suporte para ganchos do GitLab, adiciona um webhook de projeto com destino a URL `/hooks/gitlab/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitlab/`).

Veja também:

POST /hooks/gitlab/, Acessando repositórios do Hosted Weblate

Receber alterações do Pagure automaticamente

Novo na versão 3.3.

O Weblate tem suporte para ganchos Pagure. Adicione um webhook com destino a URL `/hooks/pagure/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/pagure/`). Isso pode ser feito em *Web-hooks* em *Project options*:

The screenshot shows the Weblate interface for a project named 'nijel-test'. The top navigation bar includes the 'fedora PAGURE' logo, 'Browse', 'Create', and a user profile icon. Below this, the project name 'nijel-test' is displayed alongside buttons for 'New Issue', 'Open PR', 'Fork', and 'Clone'. A secondary navigation bar contains links for 'Source', 'Issues' (0), 'Pull Requests' (0), 'Stats', and 'Settings' (which is highlighted).

The 'Settings' page is divided into two main sections. On the left is a sidebar menu with the following options: 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 main content area is titled 'Project Options' and contains several configuration items:

- ☐ Activate always merge
- ☐ Activate disable non fast-forward merges
- ☐ Activate Enforce signed-off commits in pull-request
- ☒ Activate fedmsg notifications
- ☒ Activate Issue tracker
- ☐ Activate Issue tracker read only
- ☐ Activate Issues default to private
- Activate Minimum score to merge pull-request:
- ☐ Activate notify on commit flag
- ☐ Activate notify on pull-request flag
- ☐ Activate Only assignee can merge pull-request
- ☐ Activate open metadata access to all
- ☐ Activate project documentation
- ☐ Activate pull request access only
- ☒ Activate pull requests
- ☒ Activate stomp notifications

Below these options, there is a section for 'Activate Web-hooks' with a text input field containing 'https://hosted.weblate.org/hooks/pagure/'. Below the input field are two buttons: 'Update' and 'Test web-hook'.

At the bottom of the 'Project Options' section, there is a link 'Learn more about' followed by a list of links: 'Flags', 'Tracker read-only', 'Pull-request access only', 'Roadmap on Issue page', and 'fedmsg notifications'.

Veja também:

POST /hooks/pagure/, Acessando repositórios do Hosted Weblate

Receber alterações dos Azure Repos automaticamente

Novo na versão 3.8.

O Weblate tem suporte para webhooks dos Azure Repos, adicione um webhook para o evento *Code pushed* com destino para URL `/ganchos/azure/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/azure/`). Isso pode ser feito em *Service hooks* em *Project settings*.

Veja também:

Webhooks no manual do Azure DevOps, *POST /hooks/azure/, Acessando repositórios do Hosted Weblate*

Receber alterações dos Gitea Repos automaticamente

Novo na versão 3.9.

Weblate tem suporte para webhooks do Gitea, adicione um *Gitea Webhook* para o evento *Push events* com destino ao URL `/hooks/gitea/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitea/`). Isso pode ser feito no *Webhooks* em *Settings* do repositório.

Veja também:

Webhooks no manual do Gitea, *POST /hooks/gitea/*, *Acessando repositórios do Hosted Weblate*

Receber alterações de Gitee Repos automaticamente

Novo na versão 3.9.

O Weblate tem suporte para webhooks Gitee, adicione um *WebHook* para o evento *Push* com destino para URL `/hooks/gitee/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitee/`). Isso pode ser feito em *WebHooks* sob *Management* do repositório.

Veja também:

Webhooks no manual do Gitee, *POST /hooks/gitee/*, *Acessando repositórios do Hosted Weblate*

Atualizar repositórios *nightly* automaticamente

O Weblate busca automaticamente repositórios remotos *nightly* para melhorar o desempenho ao mesclar alterações mais tarde. Pode opcionalmente transformar isso em fazer mesclagens noturnas também, ativando *AUTO_UPDATE*.

2.9.2 Fazendo push das alterações do Weblate

Cada componente de tradução pode ter uma URL de push configurada (veja *URL de submissão do repositório*) e, nesse caso, o Weblate será capaz de fazer push da alteração ao repositório remoto. O Weblate também pode ser configurado para fazer push automaticamente das alterações em cada commit (isso é a predefinição, veja *Enviar ao submeter*). Se não quiser que seja feito push automático das alterações, pode fazer-lo manualmente em *Manutenção do repositório* ou a usar API via *wlc push*.

As opções de push diferem com base no *Integração de controlo de versões* usado, mais detalhes são encontrados nesse capítulo.

No caso de não querer pushes diretos pelo Weblate, há suporte para pull requests do *GitHub*, do *GitLab* e do *Pagure* ou revisões do *Gerrit*, pode ativá-los a escolher *GitHub*, *GitLab*, *Gerrit* ou *Pagure*, como *Sistema de controlo de versões* em *Configuração de componente*.

No geral, as opções a seguir estão disponíveis com Git, GitHub e GitLab:

Configuração desejada	Sistema de controlo de versões	URL de submissão do repositório	Ramo do push
Sem push	<i>Git</i>	vazio	vazio
Push diretamente	<i>Git</i>	URL de SSH	vazio
Empurrar para um ramo separado	<i>Git</i>	URL de SSH	Nome do ramo
Pull request de GitHub do fork	<i>GitHub</i>	vazio	vazio
Pull request de GitHub do ramo	<i>GitHub</i>	URL de SSH ¹	Nome do ramo
Merge request de GitLab do fork	<i>GitLab</i>	vazio	vazio
Merge request de GitLab do ramo	<i>GitLab</i>	URL de SSH ^{Página 62, 1}	Nome do ramo
Merge request de Pagure do fork	<i>Pagure</i>	vazio	vazio
Merge request de Pagure do ramo	<i>Pagure</i>	URL de SSH ^{Página 62, 1}	Nome do ramo

Nota: Também pode ativar o push automático de alterações após o Weblate fazer commit, isso pode ser feito em *Enviar ao submeter*.

Veja também:

Consulte *Acessando repositórios* para configurar chaves de SSH e *Commits adiados* para obter informações sobre quando o Weblate decide fazer commit de alterações.

Ramos protegidos

Se estiver a usar o Weblate em ramo protegido, pode configurá-lo para usar pull requests e executar revisão real sobre as traduções (o que pode ser problemático para idiomas que não conhece). Uma abordagem alternativa é abrir mão desta limitação em favor do utilizador de push no Weblate.

Por exemplo, no GitHub, isso pode ser feito na configuração do repositório:

¹ Pode estar vazia caso o *Repositório do código-fonte* tenha suporte a push.

☒ **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 Mesclar ou rebase

Por predefinição, o Weblate mescla o repositório upstream para dentro do seu próprio. Esta é a maneira mais segura no caso de também acessar o repositório subjacente por outros meios. Caso não precise disso, pode permitir fazer rebase de alterações em upstream, o que produzirá um histórico com menos compromissos de mesclagem.

Nota: Rebasing pode causar problemas em caso de mesclagens complicadas, então considere cuidadosamente se quer ou não ativá-los.

2.9.4 Interagir com os outros

O Weblate facilita a interação com outras pessoas a usar a API dele.

Veja também:

API REST do Weblate

2.9.5 Commits adiados

O comportamento do Weblate é de agrupar commits do mesmo autor num só commit, se for possível. Isso reduz a quantidade de commits consideravelmente, no entanto, pode precisar de dizer explicitamente para fazer os commits no caso de querer deixar o repositório VCS em sincronia, por exemplo, para mesclarem (isso é por predefinição permitido para o grupo *Managers*, consulte [Lista de privilégios](#)).

As alterações neste modo têm o commit delas feitas assim que qualquer uma das seguintes condições são cumpridas:

- Outra pessoa altera uma cadeia já alterada.
- Um merge do upstream é feito.
- Um commit explícito é solicitado.
- A alteração é mais antiga do que o período definido como *Idade das alterações a fazer commit* em *Configuração de componente*.

Dica: Os commits são criados para cada componente. Então, caso tenha muitos componentes, ainda verá muitos commits. Pode utilizar a extensão *Squash de commits git* neste caso.

Se quiser fazer commit das alterações com mais frequência e sem verificar a idade, pode agendar uma tarefa regular para realizar um commit:

```
CELERY_BEAT_SCHEDULE = {
    # Unconditionally commit all changes every 2 minutes
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Ommiting hours will honor per component settings,
        # otherwise components with no changes older than this
        # won't be committed
        "kwargs": {"hours": 0},
        # How frequently to execute the job in seconds
        "schedule": 120,
    }
}
```

2.9.6 Processar repositório com scripts

A maneira de personalizar como o Weblate interage com o repositório é com *Extensões*. Consulte [Escrevendo scripts para extensões](#) para obter informações sobre como executar scripts externos através de extensões.

2.9.7 Manter traduções iguais entre componentes

Uma vez que tenha vários componentes de tradução, pode garantir que as mesmas cadeias tenham a mesma tradução. Isso pode ser alcançado em vários níveis.

Propagação de tradução

Com a propagação de tradução ativada (que é a predefinição, consulte *Configuração de componente*), todas as novas traduções são feitas automaticamente em todos os componentes com cadeias correspondentes. Estas traduções são devidamente creditadas ao utilizador que traduz atualmente em todos os componentes.

Nota: A propagação de tradução requer a chave para ser compatível com formatos de tradução monolíngue, por isso tenha isso em mente ao criar chaves de tradução.

Verificação de consistência

A verificação check-inconsistente é acionada sempre que as cadeias são diferentes. Pode usar isso para rever tais diferenças manualmente e escolher a tradução certa.

Tradução automática

A tradução automática com base em diferentes componentes pode ser uma maneira de sincronizar as traduções entre os componentes. Pode acioná-la manualmente (veja *Tradução automática*) ou fazê-la ser executada automaticamente na atualização do repositório a usar uma extensão (veja *Tradução automática*).

2.10 Licenciar traduções

Pode especificar sob quais traduções de licença são contribuídas. Isto é especialmente importante se as traduções forem abertas ao público, para estipular para que elas possam ser usadas.

Deve especificar as informações da licença da *Configuração de componente*. Deve evitar exigir um contrato de licença de colaborador, embora seja possível.

2.10.1 Informações de licença

Ao especificar informações de licenças (nome da licença e URL), essas informações são mostradas na secção de informações de tradução do respetivo *Configuração de componente*.

Normalmente este é o melhor lugar para publicar informações de licenciamento se nenhum consentimento explícito for necessário. Se o seu projeto ou tradução não for livre, provavelmente precisa de consentimento prévio.

2.10.2 Acordo de contribuidor

Se especificar um contrato de licença de colaborador, apenas os utilizadores que concordaram com ele poderão contribuir. Este é um passo claramente visível ao acessar a tradução:

Contribuição para esta tradução requer que você concorde com um acordo de contribuição de colaborador. [View contributor agreement](#)

Language	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
Czech 🇨🇪 GPL-3.0	✓					
Hebrew 🇮🇱 GPL-3.0	✓					
Hungarian 🇮🇪 GPL-3.0	81%	4	5			
English 🇬🇧 GPL-3.0	✓					

[Start new translation](#)

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

O texto inserido é formatado em parágrafos e ligações externas podem ser incluídas. A marcação HTML não pode ser usada.

2.10.3 Licenças de utilizador

Todos utilizadores podem rever todas as licenças de tradução de todos os projetos públicos na instância do seu perfil:

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/Djangojs](#) [WeblateOrg/Django](#) [WeblateOrg/WebateOrg](#) [WeblateOrg/Language names](#)

MIT License [MIT](#) ⓘ

[WeblateOrg/Android](#)

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

2.11 Processo de tradução

2.11.1 Votação de sugestão

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 *Configuração de componente* 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*.

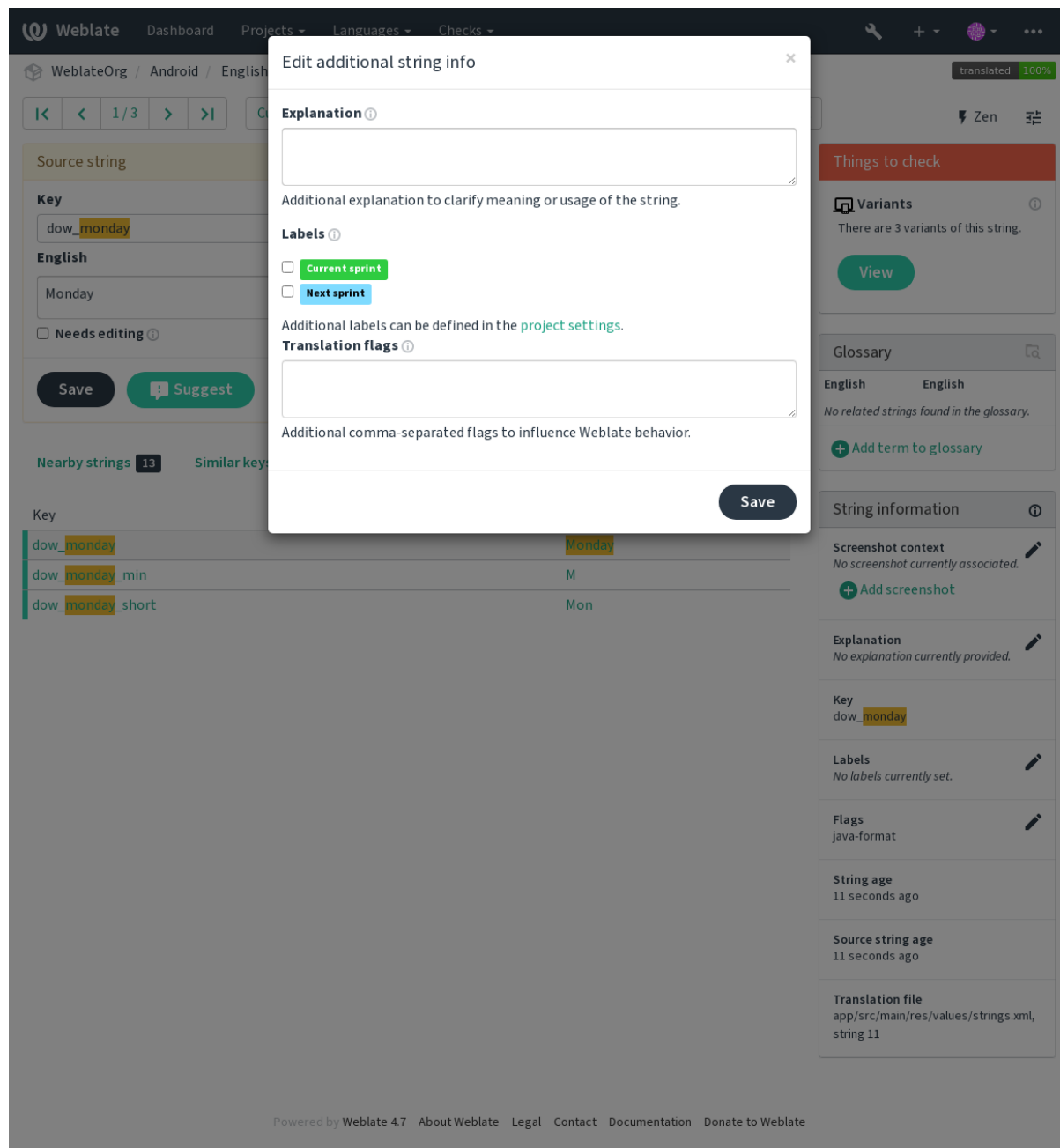
You can combine these with *access control* into one of the following setups:

- 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 Informações adicionais sobre cadeias fonte

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*.

Webate

Dashboard

Projects

Languages

Checks

WebateOrg

Django

Czech

Translate

translated 96%

11 / 26

All strings

Position and priority

Zen

Translation

Explanation

Help text for automatic translation tool

English

Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

Needs editing

Save

Suggest

Skip

Nearby strings

Comments

Automatic suggestions

Other languages

History

Context	English	Czech
	Files	Soubory
	Automatic translation	Automatický překlad
	Add new translation string	Add new translation string
	Translation status	Stav překladu
	%(count)s word	%(count)s slovo
	Other components	Další součásti
	Translation file	Soubor s překladem
	Download	Stáhnout
	Browse all translation changes	Procházet všechny změny v překladu.
	Automatic translation takes existing translations in this project and applies them to the current component. It can be used to push translations to a different branch, to fix inconsistent translations or to translate a new component using translation memory.	Automatický překlad použije stávající 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 aktivita
	Last change	Poslední změna
	Last author	Poslední autor
Question for a mathematics-based CAPTCHA, the %s is an arithmetic problem	What is %s?	Kolik to je?
	The string uses three dots (...) instead of an ellipsis character (...)	

Glossary

English

Czech

machine

strojový

weblateorg

translation

překlad

weblateorg

project

projekt

weblateorg

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

Help text for automatic translation tool

Labels

No labels currently set.

Flags

No flags currently set.

Source string location

weblate/templates/translation.html:212

String age

a second ago

Source string age

2 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 11

Powered by Weblate 4.7

About Weblate

Legal

Contact

Documentation

Donate to Weblate

2.11. Processo de tradução

265

Strings prioritization

Novo na versão 2.0.

String priority can be changed to offer higher priority strings for translation earlier by using the `priority` flag.

Dica: This can be used to order the flow of translation in a logical manner.

Veja também:

Verificações de qualidade

Marcadores de tradução

Novo na versão 2.4.

Alterado na versão 3.3: Previously called *Quality checks flags*, it no longer configures only checks.

The default set of translation flags is determined by the translation *Configuração de componente* and the translation file. However, you might want to use it to customize this per source string.

Veja também:

Verificações de qualidade, Personalizando comportamento a usar marcadores

Explicação

Alterado na versão 4.1: In previous versions this has been called *Extra context*.

Use the explanation to clarify scope or usage of the translation. You can use Markdown to include links and other markup.

Contexto visual para cadeias

Novo na versão 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.

The uploaded screenshot is shown in the translation context sidebar:

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows the translation of a string from English to Czech. The English string is 'Help text for automatic translation tool'. The Czech translation is 'Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.' The status is 'Needs editing'. Buttons for 'Save', 'Suggest', and 'Skip' are visible.

The right sidebar contains several sections:

- Glossary:** Shows terms for 'machine translation' and 'project' in both English and Czech.
- String information:** Includes a 'Screenshot context' section with a placeholder for a screenshot, an 'Add screenshot' button, and other metadata like 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.
- Translation memory:** Shows a table of previous translations with columns for Translation, Source, Origin, and Similarity. A match with 100% similarity is shown.

At the bottom, there is a footer with links: 'Powered by Weblate 4.7', 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

In addition to *Informações adicionais sobre cadeias fonte*, screenshots have a separate management interface under the *Tools* menu. Upload screenshots, assign them to source strings manually, or use optical character recognition to do so.

Once a screenshot is uploaded, this interface handles management and source string association:

Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

+

▾

...

WeblateOrg

Django

Screenshots

Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			
Screenshot is shown to add visual context for all listed source strings.			

Assign source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			

Source string search

Search

Automatically recognize

Image

Source string

Hello, world!⌵

One
Orangutan has %d banana.⌵

Other
Orangutan has %d bananas.⌵

Try Weblate at <http://demo.weblate.org/>!⌵

Thank you for using Weblate.

Screenshot is shown to add visual context for all listed source strings.

Edit screenshot

Screenshot name

Automatic translation

Image

Currently: screenshots/screenshot.png

Change:

Choose File

No file chosen

Upload JPEG or PNG images up to 2000x2000 pixels.

Save

Screenshot details

Created	now
Uploaded by	<div><div></div><div>testuser</div></div>
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

Powered by Weblate 4.7

About Weblate

Legal

Contact

Documentation

Donate to Weblate

2.12 Verificações e correções

2.12.1 Correções automáticas personalizadas

Também pode implementar a sua própria correção automática, além das predefinidas e incluí-las em `AUTO-FIX_LIST`.

As correções automáticas são poderosas, mas também podem causar danos; tenha cuidado ao escrever uma.

Por exemplo, a correção automática a seguir iria substituir cada ocorrência da cadeia `foo`, numa tradução com `bar`:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#

from django.utils.translation import gettext_lazy as _

from weblate.trans.autofixes.base import AutoFix

class ReplaceFooWithBar(AutoFix):
    """Replace foo with bar."""

    name = _("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False
```

Para instalar verificações personalizadas, forneça um caminho totalmente qualificado à classe Python em `AUTO-FIX_LIST`, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

2.12.2 Personalizando comportamento a usar marcadores

Pode ajustar o comportamento de Weblate (principalmente de verificações) para cada cadeia fonte (na revisão de cadeias fonte, veja *Informações adicionais sobre cadeias fonte*) ou em *Configuração de componente (Marcadores de tradução)*. Alguns formatos de ficheiro também permitem especificar sinalizadores diretamente no formato (veja *Formatos de ficheiros suportados*).

As etiquetas são separadas por vírgulas, os parâmetros são separados por caracteres de dois pontos. Pode usar aspas para incluir espaços em branco ou caracteres especiais na cadeia. Por exemplo:

```
placeholders:"special:value":"other value", regex:.*
```

Aqui está uma lista de sinalizadores atualmente aceites:

rst-text Trata um texto como um documento reStructuredText, afeta *Tradução inalterada*.

md-text Trata o texto como um documento de Markdown.

dos-eol Usa marcadores de ponta de linha do DOS em vez dos Unix (`\r\n` em vez de `\n`).

url A cadeia deve consistir apenas numa URL.

safe-html A cadeia deve fazer seguro para HTML, veja *HTML inseguro*.

read-only A cadeia é somente leitura e não deve ser editada no Weblate, veja *Cadeias somente leitura*.

priority:N Prioridade da cadeia. As cadeias de maior prioridade são apresentados primeiro para tradução. A prioridade predefinida é 100, quanto maior prioridade que um texto tem, mais cedo é oferecido para tradução.

max-length:N Limita o comprimento máximo de uma cadeia a N caracteres, veja *Tamanho máximo da tradução*.

xml-text Trata o texto como documento XML, afeta *Sintaxe XML* e *Markup XML*.

font-family:NOME Define a família de letras para verificações da renderização, veja *Gerir letras*.

font-weight:PESO Define o peso da letra para verificações da renderização, veja *Gerir letras*.

font-size:SIZE Define o tamanho da letra para verificações da renderização, veja *Gerir letras*.

font-spacing:ESPAÇAMENTO Define o espaçamento de letras para verificações da renderização, veja *Gerir letras*.

placeholders:NOME:NOME2:... Cadeias de espaço reservado esperados na tradução, veja *Espaços reservados*.

replacements:DE:PARA:DE2:PARA2... Substituições para realizar ao verificar parâmetros de texto resultantes (por exemplo, em *Tamanho máximo da tradução* ou *Tamanho máximo da tradução*). O caso de uso típico para isso é expandir objetos colocáveis (*placeables*) para garantir que o texto se encaixe mesmo com valores longos, por exemplo: `replacements:%s:"John Doe"`.

variants:FONTE Marca esta cadeia como uma variante do texto com a fonte correspondente. Veja *variants*.

regex:EXPRESSÃO REGULAR Expressão regular para corresponder à tradução; veja *Expressão regular*.

forbidden Indica tradução proibida num glossário, veja *Traduções proibidas*.

python-format, c-format, object-pascal-format, php-format, python-brace-format, javascript-format

Trata todos os textos como textos de formato, afeta *Formato Python*, *Formato C*, *Formato Object Pascal*, *Formato PHP*, *Formato Qt*, *Forma plural Qt*, *Formato Ruby*, *Formatação vue I18n*, *Formato de chaveta Python*, *Formato JavaScript*, *Formato C#*, *Formato Java*, *Formato de Mensagem Java*, *Formato Lua*, *Formato Scheme*, *Tradução inalterada*.

strict-same Faz com que «Tradução não alterada» evite usar a lista negra de palavras embutidas, veja *Tradução inalterada*.

check-glossary Ativa a verificação de qualidade «Não seguir glossário».

ignore-bbcode Pular a verificação de qualidade «Markup de BBcode».

ignore-duplicate Pular a verificação de qualidade «Palavras consecutivas duplicadas».

ignore-check-glossary Ignora a verificação de qualidade «Não seguir glossário».

ignore-double-space Pular a verificação de qualidade «Espaço duplo».

ignore-angularjs-format Pular a verificação de qualidade «Cadeia de interpolação AngularJS».

ignore-c-format Pular a verificação de qualidade «Formato C».

ignore-c-sharp-format Pular a verificação de qualidade «Formato C#».

ignore-es-format Pular a verificação de qualidade «Literais de modelo de ECMAScript».

ignore-i18next-interpolation Pular a verificação de qualidade «Interpolação de i18next».

ignore-java-format Pular a verificação de qualidade «Formato Java».

ignore-java-messageformat Pular a verificação de qualidade «MessageFormat do Java».

ignore-javascript-format Pular a verificação de qualidade «Formato JavaScript».

ignore-lua-format Ignora a verificação de qualidade «Formato Lua».

ignore-object-pascal-format Ignora a verificação de qualidade «Formato de Object Pascal».

ignore-percent-placeholders Pular a verificação de qualidade «Espaços reservados de percentagem».

ignore-perl-format Pular a verificação de qualidade «Formato Perl».

ignore-php-format Pular a verificação de qualidade «Formato PHP».

ignore-python-brace-format Pular a verificação de qualidade «Formato de chaves Python».

ignore-python-format Pular a verificação de qualidade «Formato Python».

ignore-qt-format Pular a verificação de qualidade «Formato Qt».

ignore-qt-plural-format Pular a verificação de qualidade «Formato de plural Qt».

ignore-ruby-format Pular a verificação de qualidade «Formato Ruby».

ignore-vue-format Pular a verificação de qualidade «Formatação Vue I18n».

ignore-translated Pular a verificação de qualidade «Foi traduzido».

ignore-inconsistent Pular a verificação de qualidade «Inconsistente».

ignore-kashida Pular a verificação de qualidade «Letra Kashida usada».

ignore-md-link Pular a verificação de qualidade «Links Markdown».

ignore-md-reflink Pular a verificação de qualidade «Referências Markdown».

ignore-md-syntax Pular a verificação de qualidade «Sintaxe Markdown».

ignore-max-length Pular a verificação de qualidade «Comprimento máximo da tradução».

ignore-max-size Pular a verificação de qualidade «Tamanho máximo da tradução».

ignore-escaped-newline Ignora a verificação de qualidade «\n não correspondente».

ignore-end-colon Ignora a verificação de qualidade «Caractere de dois pontos não correspondente».

ignore-end-ellipsis Pular a verificação de qualidade «Reticências não correspondentes».

ignore-end-exclamation Pular a verificação de qualidade «Ponto de exclamação não correspondente».

ignore-end-stop Pular a verificação de qualidade «Ponto final não correspondente».

ignore-end-question Pular a verificação de qualidade «Ponto de interrogação não correspondente».

ignore-end-semicolon Pular a verificação de qualidade «Ponto e vírgula não correspondente».

ignore-newline-count Pular a verificação de qualidade «Quebras de linha não correspondentes».

ignore-plurals Pular a verificação de qualidade «Faltam plurais».

ignore-placeholders Pular a verificação de qualidade «Espaços reservados».

ignore-punctuation-spacing Ignora a verificação de qualidade «Espaçamento de pontuação».

ignore-regex Pular a verificação de qualidade «Expressão regular».

ignore-same-plurals Pular a verificação de qualidade «Mesmos plurais».

ignore-begin-newline Pular a verificação de qualidade «Nova linha no início».

ignore-begin-space Pular a verificação de qualidade «Espaços no início».

ignore-end-newline Pular a verificação de qualidade «Nova linha no final».

ignore-end-space Pular a verificação de qualidade «Espaço no final».

ignore-same Ignora a verificação de qualidade «Tradução não alterada».

ignore-safe-html Pular a verificação de qualidade «HTML inseguro».

ignore-url Pular a verificação de qualidade «URL».

ignore-xml-tags Pular a verificação de qualidade «Marcação XML».

ignore-xml-invalid Pular a verificação de qualidade «Sintaxe XML».

ignore-zero-width-space Pular a verificação de qualidade «Espaço com largura zero».

ignore-ellipsis Pular a verificação de qualidade «Reticências».

ignore-long-untranslated Pular a verificação de qualidade «Não traduzido a muito tempo».

ignore-multiple-failures Pular a verificação de qualidade «Várias verificações com falha».

ignore-unnamed-format Pular a verificação de qualidade «Várias variáveis sem nome».

ignore-optional-plural Pular a verificação de qualidade «Não pluralizado».

Nota: Geralmente, a regra é chamada `ignore-*` para qualquer verificação, a usar o identificador dele, para que possa usá-la mesmo para as suas verificações personalizadas.

Essas etiquetas são entendidas tanto nas configurações de *Configuração de componente*, por configurações de cadeias fonte quanto no próprio ficheiro de tradução (por exemplo, no GNU gettext).

2.12.3 Forçar verificações

Novo na versão 3.11.

Pode configurar uma lista de verificações que não podem ser ignoradas a definir *Verificações impostas* em *Configuração de componente*. Cada verificação listada não pode ser ignorada na interface do utilizador e qualquer cadeia com falha nesta verificação é marcada como *Precisa de edição* (veja *Estados de tradução*).

2.12.4 Gerir letras






Novo na versão 3.7.


Dica: As fontes carregadas no Weblate são usadas puramente para fins de verificação *Tamanho máximo da tradução*, elas não têm efeito na interface de utilizador do Weblate.

A verificação *Tamanho máximo da tradução* usada para calcular as dimensões do texto renderizado precisa ser carregada no Weblate e seleccionada a usar um sinalizador de tradução (ver *Personalizando comportamento a usar marcadores*).

Ferramenta de gestão de fontes do Weblate em *Fonts* sob o menu *Gerir* do seu projeto de tradução fornece interface para carregar e gerir fontes. As fontes TrueType ou OpenType podem ser carregadas, configurar grupos de fontes e usá-los na verificação.

Os grupos de letras permitem definir letras diferentes para idiomas diferentes, o que é normalmente necessário para idiomas não-latinos:

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

 WeblateOrg / [Font groups](#) / default-font

Font group

Name	default-font		
Default font	Source Sans Pro Bold		
Japanese	language override	Droid Sans Fallback Regular	Remove
Korean	language override	Droid Sans Fallback Regular	Remove
Delete			

Add language override

Language

Font

[Save](#)

Edit font group

Font group name

default-font

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font


Source Sans Pro Bold

Default font is used unless per language override matches.

[Save](#)

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

O grupos de letras são identificados pelo nome, que não pode conter espaços ou caracteres especiais, de modo que ele pode ser facilmente utilizado na definição da verificação:

 Weblate Dashboard Projects Languages Checks

WeblateOrg / Fonts

Font groups Fonts

Group name	Default font	Language overrides	
default-font	Source Sans Pro Bold	Japanese: Droid Sans Fallback Regular Korean: Droid Sans Fallback Regular	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 4.7 About Weblate Legal Contact Documentation Donate to Weblate

A família de letras e o estilo são automaticamente reconhecidos após carregá-los:

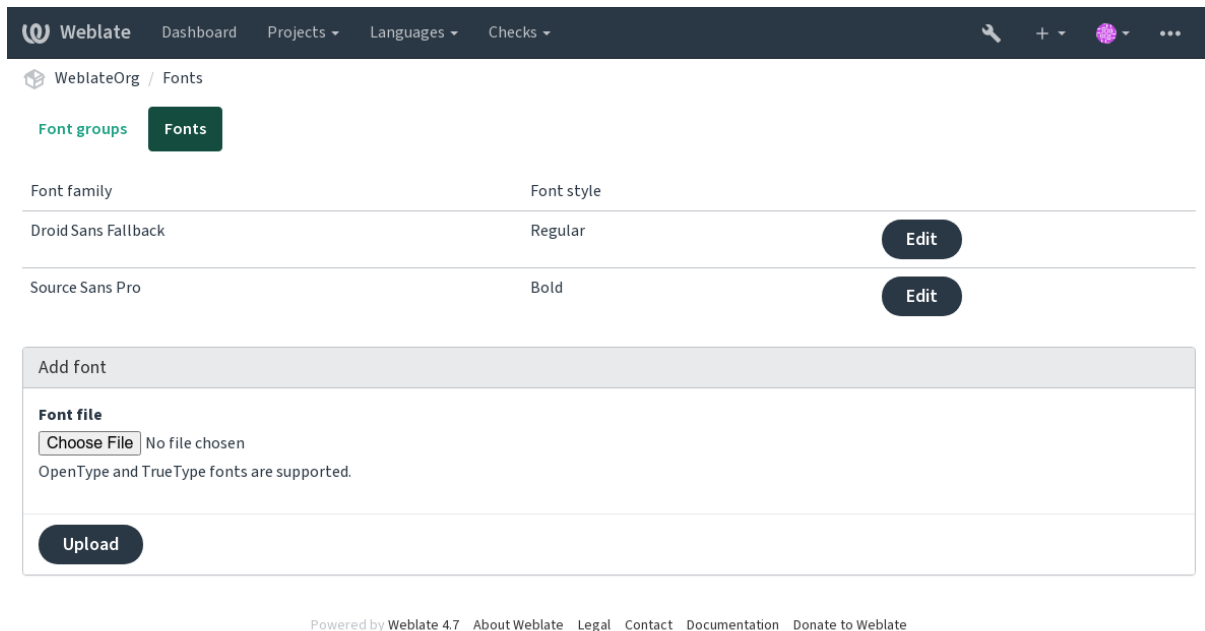
 Weblate Dashboard Projects Languages Checks

WeblateOrg / Fonts / Droid Sans Fallback Regular

Font	
Font family	Droid Sans Fallback
Font style	Regular
File size	3939852
Created	now
Uploaded by	 testuser
Used in groups	
Delete	

Powered by Weblate 4.7 About Weblate Legal Contact Documentation Donate to Weblate

Pode ter muitas letras carregadas para Weblate:



Para usar as letras para verificar o comprimento da cadeia, passe-a os sinalizadores apropriados (veja [Personalizando comportamento a usar marcadores](#)). Provavelmente precisará dos seguintes:

max-size:500 Define o máximo de largura.

font-family:ubuntu Define o grupo de letras para usar especificando seu identificador.

font-size:22 Define o tamanho da letra.

2.12.5 Escrever próprias verificações

Uma ampla gama de verificações de qualidade são incorporadas, (veja [Verificações de qualidade](#)), embora eles possam não cobrir tudo o que deseja verificar. A lista de verificações realizadas pode ser ajustada a usar [CHECK_LIST](#) e também pode adicionar verificações personalizadas.

1. Crie uma subclasse de `weblate.checks.Check`
2. Defina alguns atributos.
3. Implemente o método `check` (se quiser lidar com plurais no seu código) ou o método `check_single` (que faz isso por si).

Alguns exemplos:

Para instalar verificações personalizadas, forneça um caminho totalmente qualificado à classe Python em [CHECK_LIST](#), veja [Verificações de qualidade personalizadas, extensões e correções automáticas](#).

Verificar se o texto de tradução não contém «foo»

Esta é uma verificação bastante simples que apenas verifica se a tradução não possui a cadeia «foo».

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
```

(continues on next page)

(continuação da página anterior)

```
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Simple quality check example."""

from django.utils.translation import gettext_lazy as _

from weblate.checks.base import TargetCheck

class FooCheck(TargetCheck):

    # Used as identifier for check, should be unique
    # Has to be shorter than 50 characters
    check_id = "foo"

    # Short name used to display failing check
    name = _("Foo check")

    # Description for failing check
    description = _("Your translation is foo")

    # Real check code
    def check_single(self, source, target, unit):
        return "foo" in target
```

Verificando se os plurais de texto de tradução tcheca são diferentes

Usa as informações de idioma para verificar se as duas formas plurais no idioma tcheco não são os mesmos.

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#
"""Quality check example for Czech plurals."""

from django.utils.translation import gettext_lazy as _

from weblate.checks.base import TargetCheck
```

(continues on next page)

(continuação da página anterior)

```

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 Tradução automática

Suporte integrado para vários serviços de tradução automática e pode ser ativado pelo administrador a usar `MT_SERVICES` para cada um. Eles estão sujeitos aos seus termos de uso, portanto, certifique-se de ter permissão para usá-los como desejar.

O idioma fonte pode ser configurado na *Configuração de projeto*.

2.13.1 amaGama

Instalação especial do `tmserver` executado pelos autores do Virtaal.

Ative este serviço a adicionar `weblate.machinery.tmserver.AmagamaTranslation` a `MT_SERVICES`.

Veja também:

Installing amaGama, Amagama, Memória de tradução do amaGama

2.13.2 Apertium

Uma plataforma de tradução de máquina de software livre que fornece traduções para um conjunto limitado de idiomas.

A maneira recomendada de usar o Apertium é executar o seu próprio servidor Apertium-APy.

Ative este serviço a adicionar `weblate.machinery.apertium.ApertiumAPYTranslation` a `MT_SERVICES` e defina `MT_APERTIUM_APY`.

Veja também:

`MT_APERTIUM_APY`, Site do Apertium, Documentação do Apertium APy

2.13.3 AWS

Novo na versão 3.1.

O Amazon Translate é um serviço de tradução de máquina neural para traduzir textos de e para o inglês numa variedade de idiomas compatíveis.

1. Turn on this service by adding `weblate.machinery.aws.AWSTranslation` to `MT_SERVICES`.
2. Instale o módulo `boto3`.
3. Configure o Weblate.

Veja também:

`MT_AWS_REGION`, `MT_AWS_ACCESS_KEY_ID`, `MT_AWS_SECRET_ACCESS_KEY` Documentação do Amazon Translate

2.13.4 Tradução de máquina do Baidu API

Novo na versão 3.2.

Serviço de tradução de máquina fornecido pelo Baidu.

Este serviço usa uma API e precisa obter um ID e uma chave de API do Baidu para usá-lo.

Ative este serviço a adicionar `weblate.machinery.baidu.BaiduTranslation` a `MT_SERVICES` e defina `MT_BAIDU_ID` e `MT_BAIDU_SECRET`.

Veja também:

`MT_BAIDU_ID`, `MT_BAIDU_SECRET` API de tradução do Baidu

2.13.5 DeepL

Novo na versão 2.20.

DeepL é um serviço pago que fornece boa tradução de máquina para alguns idiomas. Precisa comprar a assinatura do *DeepL API* ou pode usar o plano legado :guilabel:`DeepL Pro (classic)`.

Ative este serviço a adicionar `weblate.machinery.deepl.DeepLTranslation` a `MT_SERVICES` e defina `MT_DEEPL_KEY`.

Dica: 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`.

Veja também:

`MT_DEEPL_KEY`, `MT_DEEPL_API_URL`, DeepL website, DeepL pricing, DeepL API documentation

2.13.6 LibreTranslate

Novo na versão 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`.

Veja também:

`MT_LIBRETRANSLATE_KEY`, `MT_LIBRETRANSLATE_API_URL`, LibreTranslate website, LibreTranslate repository, LibreTranslate mirrors

2.13.7 Glosbe

Dicionário gratuito e memória de tradução para quase todos os idiomas vivos.

O uso da API é gratuito, mas está sujeito à licença da fonte de dados usada. Existe um limite de chamadas que podem ser feitas de um IP num determinado período de tempo, para evitar abusos.

Ative este serviço a adicionar `weblate.machinery.glosbe.GlosbeTranslation` a `MT_SERVICES`.

Veja também:

Site do Glosbe

2.13.8 Google Tradutor

Serviço de tradução de máquina fornecido pelo Google.

Este serviço usa a API do Google Tradutor e precisa obter uma chave de API e ativar a cobrança no console da API do Google.

Para ativar este serviço, adicione `weblate.machinery.google.GoogleTranslation` em `MT_SERVICES` e defina `MT_GOOGLE_KEY`.

Veja também:

`MT_GOOGLE_KEY`, Documentação do Google Tradutor

2.13.9 API V3 do Google Tradutor (Avançado)

Serviço de tradução de máquina fornecido pelos serviços do Google Cloud.

Este serviço difere do anterior na forma como autentica. Para ativar o serviço, adicione `weblate.machinery.googlev3.GoogleV3Translation` para `MT_SERVICES` e defina

- `MT_GOOGLE_CREDENTIALS`
- `MT_GOOGLE_PROJECT`

Se `location` falhar, também pode precisar especificar `MT_GOOGLE_LOCATION`.

Veja também:

`MT_GOOGLE_CREDENTIALS`, `MT_GOOGLE_PROJECT`, `MT_GOOGLE_LOCATION` Documentação do Google Tradutor

2.13.10 Tradutor dos Serviços Cognitivos da Microsoft

Novo na versão 2.10.

Serviço de tradução de máquina fornecido pela Microsoft no portal do Azure como um dos Serviços Cognitivos.

Weblate implementa API V3 do Tradutor.

Para ativar este serviço, adicione `weblate.machinery.microsoft.MicrosoftCognitiveTranslation` a `MT_SERVICES` e defina `MT_MICROSOFT_COGNITIVE_KEY`.

API V2 de texto do Tradutor

A chave que usa com a API V2 do Tradutor pode ser usada com API 3.

API V3 de texto do Tradutor

Precisa registrar-se no portal do Azure e usar a chave obtida lá. Com as novas chaves do Azure, também precisa definir `MT_MICROSOFT_REGION` para a localidade do seu serviço.

Veja também:

`MT_MICROSOFT_COGNITIVE_KEY`, `MT_MICROSOFT_REGION`, Cognitive Services - Text Translation API, Microsoft Azure Portal

2.13.11 Serviço de Terminologia Microsoft

Novo na versão 2.19.

A API do Serviço de Terminologia Microsoft permite que acesse programaticamente a terminologia, as definições e os textos da interface do utilizador (UI) disponíveis no Portal de Idiomas por meio de um serviço da web.

Ative este serviço adicionando `weblate.machinery.microsoftterminology.MicrosoftTerminologyService` a `MT_SERVICES`.

Veja também:

API do Serviço de Terminologia Microsoft

2.13.12 ModernMT

Novo na versão 4.2.

Ligue este serviço adicionando `weblate.machinery.modernmt.ModernMTTranslation` para `MT_SERVICES` e configure `MT_MODERNMT_KEY`.

Veja também:

ModernMT API, `MT_MODERNMT_KEY`, `MT_MODERNMT_URL`

2.13.13 MyMemory

Grande memória de tradução com tradução de máquina.

O uso gratuito e anônimo está atualmente limitado a 100 solicitações/dia ou a 1000 solicitações/dia quando fornece um endereço de e-mail de contato em `MT_MYMEMORY_EMAIL`. Também pode pedir mais.

Ative este serviço adicionando `weblate.machinery.mymemory.MyMemoryTranslation` em `MT_SERVICES` e defina `MT_MYMEMORY_EMAIL`.

Veja também:

`MT_MYMEMORY_EMAIL`, `MT_MYMEMORY_USER`, `MT_MYMEMORY_KEY`, [site do MyMemory](#)

2.13.14 NetEase Sight API machine translation

Novo na versão 3.3.

Machine translation service provided by NetEase.

This service uses an API, and you need to obtain key and secret from NetEase.

Turn on this service by adding `weblate.machinery.youdao.NeteaseSightTranslation` to `MT_SERVICES` and set `MT_NETEASE_KEY` and `MT_NETEASE_SECRET`.

Veja também:

`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. First you will want to import some data to the translation memory:
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. Start tmserver to listen to your requests:

```
tmserver -d /var/lib/tm/db
```

4. Configure Weblate to talk to it:

```
MT_TMSERVER = "http://localhost:8888/tmserver/"
```

Veja também:

`MT_TMSERVER`, [tmserver](#) [Installing amaGama](#), [Amagama](#), [Amagama Translation Memory](#)

2.13.16 Yandex Translate

Machine translation service provided by 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`.

Veja também:

`MT_YANDEX_KEY`, [Yandex Translate API](#), [Powered by Yandex.Translate](#)

2.13.17 Youdao Zhiyun API machine translation

Novo na versão 3.2.

Machine translation service provided by Youdao.

This service uses an API, and you need to obtain an ID and an API key from Youdao.

Turn on this service by adding `weblate.machinery.youdao.YoudaoTranslation` to `MT_SERVICES` and set `MT_YOUDAO_ID` and `MT_YOUDAO_SECRET`.

Veja também:

`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

Novo na versão 2.20.

The *Memória de Tradução* 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`.

Veja também:

`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 Custom machine translation

You can also implement your own machine translation services using a few lines of Python code. This example implements machine translation in a fixed list of languages using dictionary Python module:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# 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 Extensões

Novo na versão 2.19.

Extensões fornecem maneiras para personalizar e automatizar o fluxo de trabalho de tradução. Administradores podem gerir extensões a partir do menu *Gerir* ↓ *Extensões* dos respectivos componente de tradução.

Dica: You can also configure add-ons using [API](#), `DEFAULT_ADDONS`, or `install_addon`.



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 ⓘ
 Bulk edit flags, labels, or states of strings.


Install


Statistics generator ⓘ
 Generates a file containing detailed info about the translation status.


Install


Pseudolocale generation ⓘ
 Generates a translation by adding prefix and suffix to source strings automatically.


Install


Contributors in comment ⓘ
 Updates the comment part of the PO file header to include contributor names and years of contributions.


Install


Customize gettext output ⓘ
 Allows customization of gettext output behavior, for example line wrapping.


Install


Generate MO files ⓘ
 Automatically generates a MO file for every changed PO file.

Install



Update PO files to match POT (msgmerge) ⓘ
 Updates all PO files (as configured by "Filemask") to match the POT file (as configured by "Template for new translations") using msgmerge.

Install


Squash Git commits ⓘ
 Squash Git commits prior to pushing changes.


repository wide

Install


Stale comment removal ⓘ
 Set a timeframe for removal of comments.

project wide

Install


Stale suggestion removal ⓘ
 Set a timeframe for removal of suggestions.

project wide

Install

Some add-ons will ask for additional configuration during installation.

2.14.1 Extensões embutidas

Tradução automática

Novo na versão 3.9.

ID da extensão weblate.autotranslate.autotranslate

Configuração

mode	Modo de tradução automático	
filter_type	Filtro de procura	
auto_source	Fonte da tradução automática	
component	Componentes	Ativar a contribuição para a memória de tradução compartilhada para que o projeto tenha acesso a componentes adicionais.
engines	Motores de tradução automática	
threshold	Limite de pontuação	

Traduz automaticamente as cadeias utilizando a tradução automática ou outros componentes.

Ela é acionada:

- Quando novas cadeias aparecem num componente.
- Uma vez por mês para cada componente, isso pode ser configurado a usar [BACKGROUND_TASKS](#).

Veja também:

Tradução automática, Manter traduções iguais entre componentes

CDN de localização JavaScript

Novo na versão 4.2.

ID da extensão weblate.cdn.cdnjs

Configuração

threshold	Limiar de tradução	Limiar para inclusão de traduções.
css_selector	Seletor de CSS	Seletor CSS para detectar elementos localizáveis.
cookie_name	Nome do cookie do idioma	Nome do cookie que armazena a preferência de idioma.
files	Extrair cadeias de ficheiros de HTML	Lista de nomes de ficheiros no repositório atual ou URLs remotas a serem analisados para obter cadeias traduzíveis.

Publica traduções na rede de distribuição de conteúdo para o uso na localização de JavaScript ou HTML.

Pode ser usado para localizar páginas HTML estáticas ou para carregar a localização no código JavaScript.

Gera um URL exclusivo para o seu componente, que pode ser incluído em páginas HTML para localizá-los. Veja weblate-cdn para mais detalhes.

Veja também:

cdn-addon-config, weblate-cdn, cdn-addon-extract, cdn-addon-html

Remover cadeias em branco

Novo na versão 4.4.

ID da extensão `weblate.cleanup.blank`

Configuração *Esta extensão não tem configuração.*

Remove cadeias não traduzidas dos ficheiros de tradução.

Use isso para não ter cadeias vazias em ficheiros de tradução (por exemplo, se a sua biblioteca de localização os exhibe como em falta em vez de recorrer ao texto fonte).

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Limpeza de ficheiros de tradução

ID da extensão `weblate.cleanup.generic`

Configuração *Esta extensão não tem configuração.*

Atualize todos os ficheiros de tradução para coincidirem com o ficheiro monolíngue base. Para a maioria dos formatos de ficheiro, significa remover as chaves de tradução obsoletas que já não existem no ficheiro base.

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Adiciona idiomas em falta

ID da extensão `weblate.consistency.languages`

Configuração *Esta extensão não tem configuração.*

Garante que um conjunto consistente de idiomas seja usado para todos os componentes de um projeto.

Os idiomas ausentes são verificados uma vez a cada 24 horas e quando novos idiomas são adicionados no Weblate.

Ao contrário da maioria dos outros, esta extensão afeta todo o projeto.

Dica: Traduza as cadeias recém-adicionadas automaticamente com *Tradução automática*.

Descoberta de componentes

ID da extensão `weblate.discovery.discovery`

Configuração

match	Expressão regular para corresponder ficheiros de tradução	
file_format	Formato de ficheiro	
name_template	Personalizar nome do componente	
base_file	Definir nome do ficheiro base monolingue	Manter vazio para ficheiros de tradução bilíngue.
new_base_file	Definir ficheiro base para as novas traduções	Nome do ficheiro utilizado para criar as novas traduções. Para gettext, escolher o ficheiro «.pot».
language_regex	Filtro de idioma	Expressão regular utilizada para filtrar ficheiros de tradução quando procurar pela máscara de ficheiro.
copy_addons	Clonar extensões do componente principal para os recém-criados	
remove	Remover componentes para os ficheiros inexistentes	
confirm	Confirmando que as correspondências acima parecem correctas	

Adiciona ou remove automaticamente componentes do projeto com base em alterações de ficheiros no sistema de controlo de versão.

Acionada toda vez que o VCS é atualizado, de outra forma semelhante ao comando de gestão `import_project`. Desta forma, pode rastrear vários componentes de tradução dentro de um VCS.

A correspondência é feita a usar expressões regulares que permitem uma configuração complexa, mas é necessário algum conhecimento para fazê-la. Alguns exemplos para casos de uso comum encontram-se na secção de ajuda de extensões.

Uma vez que acertar *Gravar*, uma prévia dos componentes correspondentes será apresentada, de onde pode verificar se a configuração realmente corresponde às suas necessidades:

W

Weblate

Dashboard

Projects

Languages

Checks

Web

l

ate

Org

Language names

Add-ons

Component discovery

Configure add-on

Please review and confirm the matched components.

Component	Matched files
Following components would be created	
Djangojs	weblate/locale/ru/LC_MESSAGES/djangojs.po (ru) weblate/locale/he/LC_MESSAGES/djangojs.po (he) weblate/locale/cs/LC_MESSAGES/djangojs.po (cs)
Django	weblate/locale/cs/LC_MESSAGES/django.po (cs) weblate/locale/he/LC_MESSAGES/django.po (he) weblate/locale/ru/LC_MESSAGES/django.po (ru)

☐ I confirm the above matches look correct

Regular expression to match translation files against

weblate/locale/(?P<language>[^\s]*)/LC_MESSAGES/(?P<component>[^\s]*)\.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|ru)\$

Regular expression to filter translation files against when scanning for filemask.

☒ Clone addons from the main component to the newly created ones

☐ Remove components for inexistant files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description
(?P<language>[^\s.]*)(/(?P<component>[^\s]*)\.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>[^\s.]*)/LC_MESSAGES/(?P<component>[^\s]*)\.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>[^\s]*)\. (?P<language>[^\s.]*)\.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>[^\s.]*)(/(?P<component>[^\s]*)/(?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>[^\s.]*)/strings-(?P<component>[^\s]*)\.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

Dica: A extensão de descoberta de componentes usa *URLs internas do Weblate*. É uma maneira conveniente de compartilhar a configuração do VCS entre vários componentes. Os componentes vinculados usam o repositório local do componente principal configurado a preencher `weblate://project/main-component` no campo *Repositório do código-fonte* (em *Gerir* ↓ *Configurações* ↓ *Sistema de controle de versão*) de cada componente. Isso economiza tempo com configuração e recursos do sistema também.

Veja também:

Template markup

Edição em massa

Novo na versão 3.11.

ID da extensão `weblate.flags.bulk`

Configuração

<code>q</code>	Consulta	
<code>state</code>	Estado a definir	
<code>add_flags</code>	Bandeiras de tradução para adicionar	
<code>remove_flags</code>	Bandeiras de tradução para remover	
<code>add_labels</code>	Etiqueta a adicionar	
<code>remove_labels</code>	Etiquetas a remover	

Edição em série de marcadores, etiquetas, ou estados de cadeias.

Automatize o etiquetação a começar com a consulta de pesquisa `NOT has:label` e adicione etiquetas até que todas as cadeias tenham todas as etiquetas necessárias. Outras operações automatizadas para metadados Weblate também podem ser feitas.

Exemplos::

Table 4: Rotula novas cadeias automaticamente

Consulta de pesquisa	<code>NOT has:label</code>
Etiqueta a adicionar	<code>recente</code>

Table 5: Marcar como somente leitura todas as entradas do changelog de Ficheiros de metadados da App Store

Consulta de pesquisa	<code>language:en AND key:changelogs/</code>
Bandeiras de tradução para adicionar	<code>read-only</code>

Veja também:

Edição em massa, Personalizando comportamento a usar marcadores, labels

Marcar as traduções inalteradas como «Precisa de edição»

Novo na versão 3.1.

ID da extensão `weblate.flags.same_edit`

Configuração *Esta extensão não tem configuração.*

Sempre que uma nova cadeia traduzível é importada de VCS e corresponde a uma cadeia fonte, esta é marcada como precisa de edição no Weblate. Isto é especialmente útil para os formatos de ficheiro que incluem cadeias não traduzidas.

Dica: Também pode reforçar a verificação *Tradução inalterada* a adicionar o marcador `strict-same` aos *Marcadores de tradução*.

Veja também:

Estados de tradução

Marcar as novas cadeias fonte como «Precisa de edição»

ID da extensão `weblate.flags.source_edit`

Configuração *Esta extensão não tem configuração.*

Sempre que uma nova cadeia é importada de VCS, esta é marcada como precisa de edição no Weblate. Deste modo pode filtrar e editar facilmente as cadeias fonte escritas pelos programadores.

Veja também:

Estados de tradução

Marcar as novas traduções como «Precisa de edição»

ID da extensão `weblate.flags.target_edit`

Configuração *Esta extensão não tem configuração.*

Sempre que uma nova cadeia de tradução é importada de VCS, esta é marcada como precisa de edição no Weblate. Deste modo pode filtrar e editar facilmente as traduções criadas pelos programadores.

Veja também:

Estados de tradução

Gerador de estatísticas

ID da extensão `weblate.generate.generate`

Configuração

<code>filename</code>	Nome do ficheiro gerado	
<code>template</code>	Conteúdo do ficheiro gerado	

Gera um ficheiro que contém a informação detalhada sobre o estado da tradução.

Pode usar um modelo do Django, tanto de nome de ficheiro e conteúdo, veja [Markdown](#) para uma descrição detalhada de marcação.

Por exemplo, a geração de um ficheiro de resumo para cada tradução:

Nome do ficheiro gerado `locale/{{ language_code }}.json`

Conteúdo

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}",
}
```

Veja também:

Template markup

Geração da pseudolocalidade

ID da extensão `weblate.generate.pseudolocale`

Configuração

source	Cadeias fonte	
target	Tradução de destino	
prefix	Prefixo da cadeia	
suffix	Sufixo da cadeia	

Gera uma tradução a adicionar um prefixo e sufixo às cadeias de origem automaticamente.

Pseudolocalidades são úteis para encontrar textos que não estão preparadas para localização. Isso é feito a alterar todas as cadeias fonte traduzíveis para facilitar a descoberta de textos inalterados ao executar a aplicação no idioma da pseudolocalidade.

Também é possível encontrar cadeias cujas contrapartes localizadas podem não se ajustar ao layout.

Dica: Pode usar idiomas reais para teste, mas existem pseudolocalidades disponíveis no Weblate - *en_XA* e *ar_XB*.

Dica: 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.

Contribuintes em comentários

ID da extensão `weblate.gettext.authors`

Configuração *Esta extensão não tem configuração.*

Atualiza a parte do comentário no cabeçalho do ficheiro PO para incluir nomes de colaboradores e anos de contribuições.

O cabeçalho do ficheiro PO terá a seguinte aparência:

```
# 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.
```

Atualizar a variável ALL_LINGUAS no ficheiro «configure»

ID da extensão `weblate.gettext.configure`

Configuração *Esta extensão não tem configuração.*

Atualiza a variável ALL_LINGUAS em ficheiros `configure`, `configure.in` ou `configure.ac`, quando uma nova tradução é adicionada.

Personalizar a saída gettext

ID da extensão `weblate.gettext.customize`

Configuração

<code>width</code>	Ajuste de linhas longas	Por predefinição, o gettext quebra as linhas a 77 caracteres e novas linhas. Com o parâmetro <code>--no-wrap</code> , quebra apenas em novas linhas.
--------------------	-------------------------	--

Permite personalizar o comportamento da saída gettext, por exemplo, a quebra de linhas.

Oferece as seguintes opções:

- Quebrar linhas em 77 caracteres e em novas linhas
- Quebrar as linhas apenas nas novas linhas
- Sem quebra de linhas

Nota: Por padrão, o gettext quebra as linhas em 77 caracteres e em novas linhas. Com o parâmetro `--no-wrap`, a quebra só é feita em novas linhas.

Atualizar ficheiro LINGUAS

ID da extensão `weblate.gettext.linguas`

Configuração *Esta extensão não tem configuração.*

Atualiza o ficheiro LINGUAS quando é adicionada uma nova tradução.

Gerar ficheiros MO

ID da extensão `weblate.gettext.mo`

Configuração

<code>path</code>	Caminho do ficheiro MO gerado	Se não especificado, será utilizada a localização do ficheiro PO.
-------------------	-------------------------------	---

Gera automaticamente um ficheiro MO para cada ficheiro PO alterado.

O local do ficheiro MO gerado pode ser personalizada e o campo para ele usa *Template markup*.

Atualizar ficheiros PO para coincidir com POT (msgmerge)

ID da extensão `weblate.gettext.msgmerge`

Configuração

<code>previous</code>	Manter «msgids» anteriores das cadeias traduzidas	
<code>no_location</code>	Remover localizações das cadeias traduzidas	
<code>fuzzy</code>	Utilizar correspondência de imprecisa	

Updates all PO files (as configured by *Máscara de ficheiros*) to match the POT file (as configured by *Modelo para novas traduções*) using **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.

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Squash de commits git

ID da extensão `weblate.git.squash`

Configuração

<code>squash</code>	Submeter squash	
<code>append_trailers</code>	Anexar linhas finalizadoras à mensagem do commit de squash	As linhas finalizadoras são linhas semelhantes aos cabeçalhos de e-mail RFC 822, no final de uma mensagem de commit de outra forma livre, como «Co-authored-by: ...».
<code>commit_message</code>	Commit message	Esta mensagem de commit será usada em vez das mensagens de commit combinadas dos commits após squash.

Comprimir as submissões Git antes de enviar as alterações.

Commits Git podem ter squash feito antes de fazer o push das alterações num dos seguintes modos:

Novo na versão 3.4.

- Todos os commits num só
- Por idioma
- Por ficheiro

Novo na versão 3.5.

- Por autor

As mensagens de commit originais são mantidas, mas a autoria é perdida a menos que *Por autor* seja seleccionada ou a mensagem de commit, seja personalizada para incluí-la.

Novo na versão 4.1.

As mensagens de commit originais podem opcionalmente ser substituídas por uma mensagem de commit personalizada.

Linhas finalizadoras (linhas de commits como `Co-authored-by: ...`) podem opcionalmente ser removidas das mensagens de commit originais e anexadas ao final da mensagem de compromisso após um squash. Isso também gera crédito próprio `Co-authored-by:` para cada tradutor.

Personalizar a saída JSON

ID da extensão `weblate.json.customize`

Configuração

<code>sort_keys</code>	Ordenar chaves JSON	
<code>indent</code>	Indentação JSON	

Permite ajustar o comportamento da saída JSON, por exemplo, a indentação ou a ordenação.

Formata as propriedades do ficheiro Java

ID da extensão `weblate.properties.sort`

Configuração *Esta extensão não tem configuração.*

Ordena o ficheiro de propriedades Java.

Purga de comentários obsoletos

Novo na versão 3.7.

ID da extensão `weblate.removal.comments`

Configuração

<code>age</code>	Dias para manter	
------------------	------------------	--

Definir um prazo para a remoção de comentários.

Isso pode ser útil para remover comentários antigos que podem ter ficado desatualizados. Use com cuidado, pois comentários ficarem velhos não significam que eles perderam importância.

Purga de sugestões obsoletas

Novo na versão 3.7.

ID da extensão `weblate.removal.suggestions`

Configuração

<code>age</code>	Dias para manter	
<code>votes</code>	Limites de votação	Limite para a remoção. Este campo não tem efeito com os votos desativados.

Definir um prazo para a remoção de sugestões.

Pode ser muito útil em relação à votação em sugestão (vej *Revisão por pares*) para remover sugestões que não recebem votos positivos suficientes num determinado período de tempo.

Atualizar ficheiros RESX

Novo na versão 3.9.

ID da extensão `weblate.resx.update`

Configuração *Esta extensão não tem configuração.*

Atualize todos os ficheiros de tradução para que correspondam ao ficheiro base monolingue original. As cadeias não utilizadas são removidas e as cadeias novas são adicionadas como cópias da cadeia fonte.

Dica: Use *Limpeza de ficheiros de tradução* se só quiser remover chaves de tradução obsoletas.

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Personalizar a saída YAML

Novo na versão 3.10.2.

ID da extensão `weblate.yaml.customize`

Configuração

<code>indent</code>	Indentação de YAML	
<code>width</code>	Ajuste de linhas longas	
<code>line_break</code>	Quebras de linha	

Permite ajustar o comportamento da saída YAML, por exemplo, o comprimento de linha ou novas linhas.

2.14.2 Personalizando a lista de extensões

A lista de extensões é configurada por `WEBLATE_ADDONS`. Para adicionar outra extensão, basta incluir o nome absoluto da classe nesta configuração.

2.14.3 Escrevendo extensões

Também pode escrever as suas próprias extensões. Crie uma subclasse de `weblate.addons.base.BaseAddon` para definir os metadados da extensão e, então, implemente uma função de retorno que vai fazer o processamento.

Veja também:

Desenvolvendo extensões

2.14.4 Escrevendo scripts para extensões

Extensões também podem ser usadas para executar scripts externos. Isso costumava estar integrado no Weblate, mas agora tem que escrever algum código para embrulhar o seu script com uma extensão.

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
```

(continues on next page)

(continuação da página anterior)

```
# 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 addon executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{ language_code }.po"
```

Para instruções de instalação, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

repositório VCS para qualquer componente.

Além disso, as seguintes variáveis de ambiente estão disponíveis:

WL_VCS

Sistema de controle de versão usado.

WL_REPO

URL do repositório upstream.

WL_PATH

Caminho absoluto ao repositório VCS.

WL_BRANCH

Novo na versão 2.11.

Ramo do repositório configurado no componente atual.

WL_FILEMASK

Máscara de ficheiro para componente atual.

WL_TEMPLATE

Nome de ficheiro de modelo para traduções monolínguas (pode estar vazio).

WL_NEW_BASE

Novo na versão 2.14.

Nome do ficheiro usado para criar novas traduções (pode estar vazio).

WL_FILE_FORMAT

Formato de ficheiro usado no componente atual.

WL_LANGUAGE

Idioma da tradução processada atualmente (não disponível para ganchos de nível de componente).

WL_PREVIOUS_HEAD

HEAD anterior após atualização (disponível apenas ao executar o gancho de pós-atualização).

WL_COMPONENT_SLUG

Novo na versão 3.9.

Slug do componente usado para construir a URL.

WL_PROJECT_SLUG

Novo na versão 3.9.

Slug de projeto usado para construir a URL.

WL_COMPONENT_NAME

Novo na versão 3.9.

Nome de componente.

WL_PROJECT_NAME

Novo na versão 3.9.

Nome do projeto.

WL_COMPONENT_URL

Novo na versão 3.9.

URL do componente.

WL_ENGAGE_URL

Novo na versão 3.9.

URL de engajamento do projeto.

Veja também:

Configuração de componente

Processamento de repositório pós-atualização

Pode ser usado para atualizar ficheiros de tradução quando a fonte VCS do upstream alterar. Para conseguir isso, lembre-se que o Weblate só vê ficheiros dos quais se fez commit com o VCS, então precisa fazer commit das alterações como parte do script.

Por exemplo, com Gulp, pode fazê-lo a usar o código seguinte:

```
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Processamento pré-commit de traduções

Use o script de commit para alterar automaticamente uma tradução antes de fazer commit dela para o repositório. É passado como um parâmetro único que consiste o nome de uma tradução atual.

2.15 Memória de Tradução

Novo na versão 2.20.

Weblate comes with a built-in translation memory consisting of the following:

- Manually imported translation memory (see *User interface*).
- Automatically stored translations performed in Weblate (depending on *Translation memory scopes*).
- Automatically imported past translations.

Content in the translation memory can be applied one of two ways:

- Manually, *Sugestões automáticas* view while translating.
- Automatically, by translating strings using *Tradução automática*, or *Tradução automática* addon.

For installation tips, see *Weblate Translation Memory*, which is turned on by default.

2.15.1 Translation memory scopes

Novo na versão 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.

Imported translation memory

Importing arbitrary translation memory data using the `import_memory` command makes memory content available to all users and projects.

Per user translation memory

Stores all user translations automatically in the personal translation memory of each respective user.

Per project translation memory

All translations within a project are automatically stored in a project translation memory only available for this project.

Memória de tradução compartilhada

All translation within projects with shared translation memory turned on are stored in a shared translation memory available to all projects.

Please consider carefully whether to turn this feature on for shared Weblate installations, as it can have severe implications:

- The translations can be used by anybody else.
- This might lead to disclosing secret information.

2.15.2 Managing translation memory

User interface

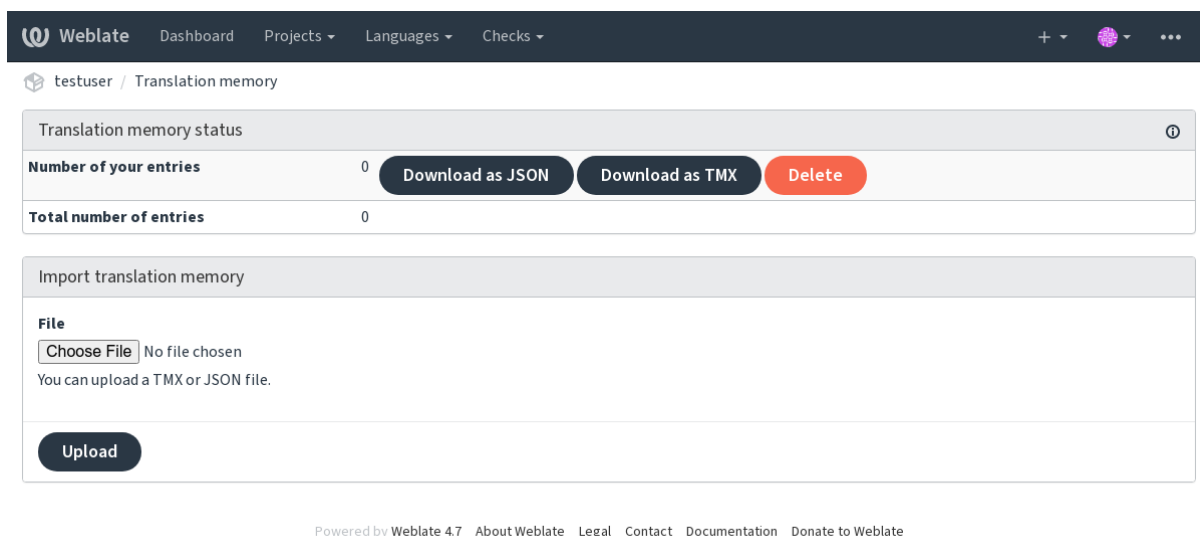
Novo na versão 3.2.

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.

Dica: Translation memory in JSON can be imported into Weblate, TMX is provided for interoperability with other tools.

Veja também:

Esquema de memória de tradução Weblate



Interface de gestão

There are several management commands to manipulate the translation memory content. These operate on the translation memory as whole, unfiltered by scopes (unless requested by parameters):

`dump_memory` Exports the memory into JSON

`import_memory` Importa ficheiros TMX ou JSON para a memória de tradução

2.16 Configuração

Todas as configurações estão armazenadas em `settings.py` (como é habitual no Django).

Nota: Após alterar qualquer uma dessas configurações, precisa reiniciar o Weblate - tanto os processos WSGI quanto os Celery.

Caso seja executado como `mod_wsgi`, precisa reiniciar o Apache para recarregar a configuração.

Veja também:

Verifique também :doc:`Django's documentation <django:ref/settings>` para parâmetros de configuração do próprio Django.

2.16.1 AKISMET_API_KEY

O Weblate pode usar o Akismet para procurar sugestões recebidas anonimamente por spam. Visite akismet.com para comprar uma chave API e associá-la a um site.

2.16.2 ANONYMOUS_USER_NAME

O nome de utilizadores não autenticados.

Veja também:

Controlo de acesso

2.16.3 AUDITLOG_EXPIRY

Novo na versão 3.6.

Quantos dias o Weblate deve manter registos de auditoria, que contêm informações sobre a atividade da conta.

A predefinição é de 180 dias.

2.16.4 AUTH_LOCK_ATTEMPTS

Novo na versão 2.14.

Quantidade máxima de tentativas de autenticação que falharam antes da aplicação da limitação de taxa.

Atualmente, isto é aplicado nos locais seguintes:

- Logins. Apaga a palavra-passe da conta, impedindo que o utilizador entre sem solicitar uma nova palavra-passe.
- Redefinições de palavra-passe. Impede que novos e-mails sejam enviados, evitando o envio de spam aos utilizadores com muitas tentativas de redefinição de palavra-passe.

A predefinição é 10.

Veja também:

Limitação de taxa,

2.16.5 AUTO_UPDATE

Novo na versão 3.2.

Alterado na versão 3.11: A opção original de ligar/desligar foi alterada para diferenciar quais cadeias são aceites.

Atualiza todos repositórios diariamente.

Dica: Útil se não estiver a usar *Hooks de notificação* para atualizar os repositórios do Weblate automaticamente.

Nota: Existem opções de ligar/desligar, além da seleção de cadeias para compatibilidade com versões anteriores.

As opções são:

"none" Sem atualizações diárias.

"remote" e também False Atualizar apenas os repositórios remotos.

"full" e também True Atualizar repositórios remotos e mesclar a cópia de trabalho.

Nota: Isto requer que *Tarefas de fundo a usar o Celery* esteja a funcionar e entrará em vigor após ser reiniciado.

2.16.6 AVATAR_URL_PREFIX

Prefixo para construção de URLs de avatars: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Os serviços seguintes funcionam:

Gravatar (predefinição), conforme <https://gravatar.com/> `AVATAR_URL_PREFIX = 'https://www.gravatar.com/'`

Libravatar, conforme <https://www.libravatar.org/> `AVATAR_URL_PREFIX = 'https://www.libravatar.org/'`

Veja também:

*Cache de avatares, `ENABLE_AVATARS`, *Avatars**

2.16.7 AUTH_TOKEN_VALID

Novo na versão 2.14.

Por quanto tempo o token de autenticação e a palavra-passe temporária dos e-mails de redefinição de palavra-passe são válidos. Definido em número de segundos, usando 172800 (2 dias) como predefinição.

2.16.8 AUTH_PASSWORD_DAYS

Novo na versão 2.15.

Quantos dias a usar a mesma palavra-passe deve ser permitido.

Nota: Mudanças de palavra-passe feitas anteriormente ao Weblate 2.15 não serão consideradas para essa política.

A predefinição é de 180 dias.

2.16.9 AUTOFIX_LIST

Lista de correções automáticas para aplicar ao gravar uma cadeia.

Nota: Forneça um caminho totalmente qualificado à classe Python que implementa a interface de correção automática.

Correções disponíveis:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace` Corresponde o espaço em branco no início e no fim da cadeia com a fonte.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis` Substitui pontos ao final (...) se a cadeia fonte tiver um caractere de reticências (...) correspondente.

`weblate.trans.autofixes.chars.RemoveZeroSpace` Remove caracteres de espaço de largura zero se a fonte não contiver nenhum.

`weblate.trans.autofixes.chars.RemoveControlChars` Remove caracteres de controle se a fonte não contiver nenhum.

weblate.trans.autofixes.html.BleachHTML Remove a marcação HTML insegura das cadeias sinalizadas como `safe-html` (veja *HTML inseguro*).

Pode selecionar quais usar:

```
AUTOFIX_LIST = (
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
)
```

Veja também:

Correções automáticas, Correções automáticas personalizadas

2.16.10 BACKGROUND_TASKS

Novo na versão 4.5.2.

Define com que frequência as tarefas de manutenção demoradas devem ser acionadas para um componente.

No momento, isso controla:

- Extensão *Tradução automática*
- Recálculo de *Verificações e correções*

Escolhas possíveis:

- `monthly` (este é o padrão)
- `weekly`
- `daily`
- `never`

Nota: O aumento da frequência não é recomendado quando o Weblate contém milhares de componentes.

2.16.11 BASE_DIR

Diretório base onde as fontes do Weblate estão localizadas. Usado para derivar vários outros caminhos por predefinição:

- `DATA_DIR`

Valor predefinido: Diretório de nível superior de fontes do Weblate.

2.16.12 BASIC_LANGUAGES

Novo na versão 4.4.

Lista de idiomas para oferecer aos utilizadores para iniciar uma nova tradução. Quando não especificado, a lista embutida é usada, o que inclui todos os idiomas comumente usados, mas sem variantes específicas de país.

Isso só limita utilizadores não privilegiados a adicionar idiomas indesejados. Os administradores do projeto ainda são apresentados com seleção completa de idiomas definidos no Weblate.

Nota: Isso não define novos idiomas para Weblate, ele filtra apenas os existentes no banco de dados.

Exemplo:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Veja também:

Definições de idioma

2.16.13 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Personaliza o cabeçalho Content-Security-Policy para Weblate. O cabeçalho é gerado automaticamente com base em integrações ativadas com serviços de terceiros (Matomo, Google Analytics, Sentry, ...).

Todos esses tem uma lista vazia como predefinição.

Exemplo:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Veja também:

Política de segurança de conteúdo, Content Security Policy (CSP)

2.16.14 CHECK_LIST

Lista de verificações de qualidade para realizar numa tradução.

Nota: Forneça um caminho totalmente qualificado à classe Python que implementa a interface de verificação.

Ajuste a lista de verificações para incluir as relevantes para si.

Todas as *Verificações de qualidade* embutidas estão ativadas por predefinição, de onde pode alterar essas configurações. Por predefinição, eles são comentados em *Amostra de configuração* para que os valores predefinidos sejam usados. Novas verificações são realizadas para cada versão nova do Weblate.

Pode desativar todas as verificações:

```
CHECK_LIST = ()
```

Pode ativar apenas algumas:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Nota: Alterar esta configuração afeta apenas as traduções recém-alteradas, as verificações existentes ainda serão armazenadas no banco de dados. Para também aplicar alterações nas traduções armazenadas, execute *update-checks*.

Veja também:

Verificações de qualidade, Personalizando comportamento a usar marcadores

2.16.15 COMMENT_CLEANUP_DAYS

Novo na versão 3.6.

Apaga comentários após uma determinada quantidade de dias. A predefinição é `None`, ou seja, nada apagado.

2.16.16 COMMIT_PENDING_HOURS

Novo na versão 2.10.

Quantidade de horas entre fazer o commit de alterações pendentes por meio da tarefa de segundo plano.

Veja também:

Configuração de componente, Idade das alterações a fazer commit, Executar tarefas de manutenção, commit_pending

2.16.17 CONTACT_FORM

Novo na versão 4.6.

Configura como o e-mail do formulário de contato está a ser enviado. Escolha uma configuração que corresponda à configuração do seu servidor de e-mail.

"reply-to" O remetente é usado como *Reply-To*, este é o comportamento padrão.

"from" O remetente é usado como *From*. O seu servidor de e-mail precisa permitir o envio de tais e-mails.

2.16.18 DATA_DIR

A pasta na qual Weblate armazena todos os dados. Ela contém ligações para repositórios VCS, um índice de texto e vários ficheiros de configuração para ferramentas externas.

Os subdiretórios seguintes geralmente existem:

home O diretório pessoal usado para invocar scripts.

ssh Chaves e configuração de SSH.

static Localização padrão para ficheiros de Django estáticos, especificados por `STATIC_ROOT`. Veja *Servir ficheiros estáticos*.

O contentor do Docker usa um volume separado para isso, veja *Volumes de contentor Docker*.

media Localização padrão para ficheiros de mídia Django, especificado por `MEDIA_ROOT`. Contém capturas de ecrã enviadas, veja *Contexto visual para cadeias*.

vcs Repositórios de controle de versão para traduções.

backups Dados de backup diário. Confira *Dados despejados para backups* para detalhes.

celery Dados do agendador Celery, consulte *Tarefas de fundo a usar o Celery*.

fonts: Fontes enviadas pelo utilizador, veja *Gerir letras*.

Nota: Este diretório tem que ser escrito pelo Weblate. Executá-lo como uWSGI significa que o utilizador `www-data` deve ter acesso de escrita.

A maneira mais fácil de conseguir isto é fazer do utilizador o proprietário do diretório:

```
sudo chown www-data:www-data -R $DATA_DIR
```

A predefinição é `$BASE_DIR/data`.

Veja também:

`BASE_DIR`, Permissões do sistema de ficheiros, Fazer backup e mover o Weblate

2.16.19 DATABASE_BACKUP

Novo na versão 3.1.

Se os backups de banco de dados devem ser armazenados como texto simples, compactado ou ignorado. Os valores autorizados são:

- `"plain"`
- `"compressed"`
- `"none"`

Veja também:

Fazer backup e mover o Weblate

2.16.20 DEFAULT_ACCESS_CONTROL

Novo na versão 3.3.

A configuração predefinida de controle de acesso para novos projetos:

0 *Público*

1 *Protegido*

100 *Privado*

200 *Personalizado*

Use *Personalizado* se está a gerir a ACL manualmente, o que significa não confiar na gestão interna do Weblate.

Veja também:

Controlo de acesso ao projeto, Controlo de acesso

2.16.21 DEFAULT_AUTO_WATCH

Novo na versão 4.5.

Configura se *Observar automaticamente projetos em contribuição* deve ser ativado para novos utilizadores. O padrão é `True`.

Veja também:

Notificações

2.16.22 DEFAULT_RESTRICTED_COMPONENT

Novo na versão 4.1.

O valor predefinido para a restrição de componentes.

Veja também:

Acesso restrito, Escopo de grupos

2.16.23 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DE- FAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DE- FAULT_MERGE_MESSAGE

Enviar mensagens predefinidas para diferentes operações, consulte *Configuração de componente* para detalhes.

Veja também:

Template markup, Configuração de componente, Mensagens de commit, add, merge e extensão

2.16.24 DEFAULT_ADDONS

Complementos predefinidos para instalar em cada componente criado.

Nota: Essa configuração afeta apenas componentes recém-criados.

Exemplo:

```
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",
    },
}
```

Veja também:

install_addon, Extensões, WEBLATE_ADDONS

2.16.25 DEFAULT_COMMITER_EMAIL

Novo na versão 2.4.

Endereço de e-mail do committer com o padrão `noreply@weblate.org`.

Veja também:

DEFAULT_COMMITER_NAME

2.16.26 DEFAULT_COMMITER_NAME

Novo na versão 2.4.

Nome do committer com o padrão `Weblate`.

Veja também:

DEFAULT_COMMITER_EMAIL

2.16.27 DEFAULT_LANGUAGE

Novo na versão 4.3.2.

Idioma fonte padrão para usar, por exemplo, em *Idioma fonte*.

O padrão é *en*. O objeto de idioma correspondente precisa existir no banco de dados.

Veja também:

Definições de idioma, Idioma fonte

2.16.28 DEFAULT_MERGE_STYLE

Novo na versão 3.4.

Mescla o estilo para quaisquer componentes novos.

- *rebase* - predefinição
- *merge*

Veja também:

Configuração de componente, Estilo de união

2.16.29 DEFAULT_SHARED_TM

Novo na versão 3.2.

Configura o valor padrão de *Utilizar memória de tradução partilhada* e `project-contrib_shared_tm`.

2.16.30 DEFAULT_TRANSLATION_PROPAGATION

Novo na versão 2.5.

Configuração predefinida para propagação de tradução, a predefinição é `True`.

Veja também:

Configuração de componente, Permitir propagação da tradução

2.16.31 DEFAULT_PULL_MESSAGE

Título para pull requests novas, a predefinição é 'Update from Weblate'.

2.16.32 ENABLE_AVATARS

Se se deve ativar avatares baseados em Gravatar para os utilizadores. Por predefinição, isto está ativado.

Avatares são buscados e armazenados em cache no servidor, diminuindo o risco de vazamento de informações privadas, acelerando a experiência do utilizador.

Veja também:

Cache de avatares, AVATAR_URL_PREFIX, Avatars

2.16.33 ENABLE_HOOKS

Se se deve ativar ganchos remotos anônimos.

Veja também:

Hooks de notificação

2.16.34 ENABLE_HTTPS

Se se deve enviar ligações para Weblate como HTTPS ou HTTP. Esta configuração afeta os e-mails enviados e as URLs absolutas geradas.

Na configuração padrão, este também é usado para várias configurações de Django relacionadas ao HTTPS - ele ativa cookies seguros, ativa/desativa HSTS ou ativa redirecionamento para URL HTTPS.

O redirecionamento HTTPS pode ser problemático em alguns casos e pode ter problemas com o redirecionamento infinito no caso de estar a usar um proxy reverso a fazer o encerramento SSL que não passa corretamente os cabeçalhos de protocolo para o Django. Ajuste a sua configuração de proxy reverso para emitir cabeçalhos X-Forwarded-Proto ou Forwarded, ou configure `SECURE_PROXY_SSL_HEADER` para permitir que o Django detecte corretamente o estado SSL.

Veja também:

`SESSION_COOKIE_SECURE,` `CSRF_COOKIE_SECURE,` `SECURE_SSL_REDIRECT,` `SECURE_PROXY_SSL_HEADER` *Definir domínio correto do site*

2.16.35 ENABLE_SHARING

Ativa/desativa o menu *Compartilhar* para que os utilizadores possam compartilhar o progresso da tradução nas redes sociais.

2.16.36 GET_HELP_URL

Novo na versão 4.5.2.

URL onde o suporte para sua instância do Weblate encontra-se.

2.16.37 GITLAB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores de GitLab.

Dica: Use isto no caso de querer que o Weblate interaja com mais deles, para um único ponto final do GitLab com *GITLAB_USERNAME* e *GITLAB_TOKEN*.

```
GITLAB_CREDENTIALS = {
  "gitlab.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitlab.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

2.16.38 GITLAB_USERNAME

O nome de utilizador GitLab para enviar merge requests para atualizações de tradução.

Veja também:

GITLAB_CREDENTIALS, *GitLab*

2.16.39 GITLAB_TOKEN

Novo na versão 4.3.

Token de acesso pessoal do GitLab usado para fazer chamadas API para atualizações de tradução.

Veja também:

GITLAB_CREDENTIALS;ref:vcs-gitlab, *GitLab*: Ficha de acesso pessoal

2.16.40 GITHUB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores GitHub.

Dica: Use isto no caso de querer que o Weblate interaja com mais deles, para um único ponto final do GitHub com *GITHUB_USERNAME* e *GITHUB_TOKEN*.

```
GITHUB_CREDENTIALS = {
  "api.github.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "github.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

2.16.41 GITHUB_USERNAME

O nome de utilizador GitHub para enviar pull request para atualizações de tradução.

Veja também:

GITHUB_CREDENTIALS, *GitHub*

2.16.42 GITHUB_TOKEN

Novo na versão 4.3.

Token de acesso pessoal GitHub usado para fazer chamadas API para enviar pull requests de tradução.

Veja também:

GITHUB_CREDENTIALS, *GitHub*, *Creating a GitHub personal access token*

2.16.43 GOOGLE_ANALYTICS_ID

ID do Google Analytics para ativar o monitoramento do Weblate a usar o Google Analytics.

2.16.44 HIDE_REPO_CREDENTIALS

Ocultar credenciais de repositório da interface web. No caso de ter URL do repositório com utilizador e palavra-passe, o Weblate irá ocultá-la quando as informações relacionadas são mostradas aos utilizadores.

Por exemplo, em vez de `https://utilizador:palavra-passe@git.example.com/repo.git`, vai mostrar apenas `“https://git.example.com/repo.git”`. Tenta limpar mensagens de erro VCS também de forma semelhante.

Nota: Isso está ativado por predefinição.

2.16.45 HIDE_VERSION

Novo na versão 4.3.1.

Esconde informação de versão de utilizadores não autenticados. Isso também faz todos as ligações de documentação apontar para a última versão, ao invés da versão de combinação da documentação instalada atualmente.

Ocultar a versão é uma prática de segurança recomendada em algumas empresas, mas não prevê invasores de descobrir a versão a sondar o comportamento.

Nota: Isto está desativado por predefinição.

2.16.46 IP_BEHIND_REVERSE_PROXY

Novo na versão 2.14.

Indica se o Weblate está a ser usado através de um proxy reverso.

Se for definido como `True`, o Weblate obtém o endereço IP de um cabeçalho definido por `IP_PROXY_HEADER`.

Aviso: Certifique-se de que está realmente a usar um proxy reverso e que ele define este cabeçalho, caso contrário, os utilizadores poderão falsificar o endereço IP.

Nota: Isto está ligado por predefinição.

Veja também:

Executar por trás de um proxy reverso, Limitação de taxa, IP_PROXY_HEADER, IP_PROXY_OFFSET

2.16.47 IP_PROXY_HEADER

Novo na versão 2.14.

Indica de qual cabeçalho o Weblate deve obter o endereço IP quando `IP_BEHIND_REVERSE_PROXY` está ativado.

A predefinição é `HTTP_X_FORWARDED_FOR`.

Veja também:

Executar por trás de um proxy reverso, Limitação de taxa, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_OFFSET

2.16.48 IP_PROXY_OFFSET

Novo na versão 2.14.

Indica qual parte de `IP_PROXY_HEADER` é usada como endereço IP do cliente.

Dependendo da configuração, este cabeçalho pode consistir em vários endereços IP (por exemplo, `X-Forwarded-For: a, b, client-ip`) e pode configurar qual endereço do cabeçalho é usado como endereço IP do cliente aqui.

Aviso: Configurar isto afeta a segurança da sua instalação, por isso deve configurá-la só para usar proxies confiáveis para determinar o endereço IP.

A predefinição é 0.

Veja também:

Executar por trás de um proxy reverso, Limitação de taxa, SECURE_PROXY_SSL_HEADER, IP_BEHIND_REVERSE_PROXY, IP_PROXY_HEADER

2.16.49 LEGAL_URL

Novo na versão 3.5.

URL onde a sua instância de Weblate mostra os documentos legais dela.

Dica: Útil se hospeda os seus documentos legais fora do Weblate para incorporá-los ao Weblate, verifique [Legal](#) para obter detalhes.

Exemplo:

```
LEGAL_URL = "https://weblate.org/terms/"
```

2.16.50 LICENSE_EXTRA

Licenças adicionais para incluir nas opções de licença.

Nota: Cada definição de licença deve ser uma tupla do seu nome curto, um nome longo e uma URL.

Por exemplo:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

2.16.51 LICENSE_FILTER

Alterado na versão 4.3: A configurar este para valor em branco desativa o alerta de licença.

Filtrar licenças da lista para mostrar. Isto também desativa o alerta de licença quando configurado para vazio.

Nota: Este filtro usa os nomes de licença curtos.

Por exemplo:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

O seguinte desativa o alerta de licença:

```
LICENSE_FILTER = set()
```

Veja também:

alerts

2.16.52 LICENSE_REQUIRED

Define se o atributo de licença em *Configuração de componente* é necessário.

Nota: Isto está desativado por predefinição.

2.16.53 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Se o comprimento de uma determinada tradução deve ser limitado. A restrição é o comprimento da cadeia fonte * 10 caracteres.

Dica: Define isto como `False` para permitir traduções mais longas (até 10.000 caracteres) independentemente do comprimento da cadeia fonte.

Nota: A predefinição é «True».

2.16.54 LOCALIZE_CDN_URL e LOCALIZE_CDN_PATH

Essas configurações definem a extensão *CDN de localização JavaScript*. `LOCALIZE_CDN_URL` define a URL raiz onde o CDN de localização está disponível e `LOCALIZE_CDN_PATH` define o caminho onde o Weblate deve armazenar ficheiros gerados que serão servidos em `LOCALIZE_CDN_URL`.

Dica: O Hosted Weblate usa o `https://weblate-cdn.com/`.

Veja também:

CDN de localização JavaScript

2.16.55 LOGIN_REQUIRED_URLS

Uma lista de URLs para as quais deseja exigir autenticação. (Além das regras predefinidas incorporadas ao Weblate).

Dica: Isto permite que proteja toda a instalação com uma palavra-passe a usar:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Dica: É desejável bloquear o acesso à API também, como mostrado no exemplo acima.

Veja também:

REQUIRE_LOGIN

2.16.56 LOGIN_REQUIRED_URLS_EXCEPTIONS

Lista de exceções para `LOGIN_REQUIRED_URLS`. Se não especificado, os utilizadores podem acessar a página de autenticação.

Algumas das exceções que pode incluir:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$", # Required for sign in
    r"/static/(.*)$", # Required for development mode
    r"/widgets/(.*)$", # Allowing public access to widgets
    r"/data/(.*)$", # Allowing public access to data exports
    r"/hooks/(.*)$", # Allowing public access to notification hooks
    r"/api/(.*)$", # Allowing access to API
    r"/js/i18n/$", # JavaScript localization
)
```

2.16.57 MATOMO_SITE_ID

ID de um site em Matomo (anteriormente Piwik) que quer rastrear.

Nota: Esta integração não suporta o Matomo Tag Manager.

Veja também:

`MATOMO_URL`

2.16.58 MATOMO_URL

URL completa (incluindo barra ao final) de uma instalação Matomo (anteriormente Piwik) que deseja usar para rastrear o uso do Weblate. Por favor, consulte <<https://matomo.org/>> para mais detalhes.

Dica: Esta integração não suporta o Matomo Tag Manager.

Por exemplo:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Veja também:

`MATOMO_SITE_ID`

2.16.59 MT_SERVICES

Alterado na versão 3.0: A configuração foi renomeada de `MACHINE_TRANSLATION_SERVICES` para `MT_SERVICES` para ser consistente com outras configurações de tradução de máquina.

Lista de serviços de tradução de máquina ativados para uso.

Nota: Muitos dos serviços precisam de configuração adicional, como chaves de API, consulte a sua documentação *Tradução automática* para mais detalhes.

```
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",  
)
```

Veja também:

Tradução automática, Sugestões automáticas

2.16.60 MT_APERTIUM_APY

URL do servidor Apertium-APy, <https://wiki.apertium.org/wiki/Apertium-apy>

Veja também:

Apertium, Tradução automática, Sugestões automáticas

2.16.61 MT_AWS_ACCESS_KEY_ID

ID da chave de acesso para Amazon Translate.

Veja também:

AWS, Tradução automática, Sugestões automáticas

2.16.62 MT_AWS_SECRET_ACCESS_KEY

Chave secreta da API para o Amazon Translate.

Veja também:

AWS, Tradução automática, Sugestões automáticas

2.16.63 MT_AWS_REGION

Nome da região para usar no Amazon Translate.

Veja também:

AWS, Tradução automática, Sugestões automáticas

2.16.64 MT_Baidu_ID

ID do cliente para a API do Baidu Zhiyun, pode registrar-se em <https://api.fanyi.baidu.com/api/trans/product/index>

Veja também:

Tradução de máquina do Baidu API, Tradução automática, Sugestões automáticas

2.16.65 MT_Baidu_SECRET

Segredo do cliente para a API do Baidu Zhiyun, pode registrar-se em <https://api.fanyi.baidu.com/api/trans/product/index>

Veja também:

Tradução de máquina do Baidu API, Tradução automática, Sugestões automáticas

2.16.66 MT_DEEPL_API_URL

Alterado na versão 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.

Anteriormente, o Weblate era classificado como uma ferramenta CAT pelo DeepL, por isso deveria usar a API v1, mas agora é entendido que deve usar a API v2. Portanto, a predefinição é v2 e pode alterá-lo para v1 no caso de ter uma assinatura CAT existente e querer que o Weblate use isso.

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.

Veja também:

DeepL, Tradução automática, Sugestões automáticas

2.16.67 MT_DEEPL_KEY

Chave de API ao API do DeepL, pode registrar-se em <https://www.deepl.com/pro.html>

Veja também:

DeepL, Tradução automática, Sugestões automáticas

2.16.68 MT_LIBRETRANSLATE_API_URL

Novo na versão 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>

Veja também:

LibreTranslate, Tradução automática, Sugestões automáticas

2.16.69 MT_LIBRETRANSLATE_KEY

Novo na versão 4.7.1.

API key for the LibreTranslate instance specified in `MT_LIBRETRANSLATE_API_URL`.

Veja também:

LibreTranslate, Tradução automática, Sugestões automáticas

2.16.70 MT_GOOGLE_KEY

Chave de API para a API v2 do Google Translate, pode registrar-se em <https://cloud.google.com/translate/docs>

Veja também:

Google Tradutor, Tradução automática, Sugestões automáticas

2.16.71 MT_GOOGLE_CREDENTIALS

Ficheiro de credenciais da API v3 do JSON obtido no console de nuvem do Google. Por favor, forneça um caminho completo do sistema operacional. As credenciais são por conta de serviço afiliada ao projeto determinado. Por favor, verifique <https://cloud.google.com/docs/authentication/getting-started> para mais detalhes.

2.16.72 MT_GOOGLE_PROJECT

ID de projeto da API v3 do Google Cloud com serviço de tradução ativado e faturamento ativado. Por favor consulte <https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project> para mais detalhes

2.16.73 MT_GOOGLE_LOCATION

A API v3 do App Engine do Google Cloud pode ser específica para uma localidade. Altere conforme o caso, se a predefinição ``global`` não lhe servir.

Consulte <https://cloud.google.com/appengine/docs/locations> para mais detalhes

Veja também:

API V3 do Google Tradutor (Avançado)

2.16.74 MT_MICROSOFT_BASE_URL

Domínio de URL base da região conforme definido na secção «URLs base».

A predefinição é `api.cognitive.microsofttranslator.com` para o Azure Global.

Para Azure China use `api.translator.azure.cn`.

2.16.75 MT_MICROSOFT_COGNITIVE_KEY

Chave do cliente para a API do Microsoft Cognitive Services Translator.

Veja também:

Tradutor dos Serviços Cognitivos da Microsoft, Tradução automática, Sugestões automáticas, Serviços Cognitivos - API de Tradução de Texto, Microsoft Azure Portal

2.16.76 MT_MICROSOFT_REGION

Prefixo da região conforme definido na secção «Autenticar com um recurso de vários serviços».

2.16.77 MT_MICROSOFT_ENDPOINT_URL

Domínio de URL de extremidade da região para token de acesso definido na secção «Autenticando com um token de acesso».

A predefinição é `api.cognitive.microsoft.com` para Azure Global.

Para Azure China, use o desfecho do Portal do Azure.

2.16.78 MT_MODERNMT_KEY

Chave API ao motor de tradução automática ModernMT.

Veja também:

ModernMT MT_MODERNMT_URL

2.16.79 MT_MODERNMT_URL

URL de ModernMT. A predefinição é `https://api.modernmt.com/` para o serviço de nuvem.

Veja também:

ModernMT MT_MODERNMT_KEY

2.16.80 MT_MYMEMORY_EMAIL

Endereço de e-mail de identificação do myMemory. Permite 1000 solicitações por dia.

Veja também:

MyMemory, Tradução automática, Sugestões automáticas, MyMemory: API technical specifications

2.16.81 MT_MYMEMORY_KEY

Chave de acesso do MyMemory para memória de tradução privada. Use-a com `MT_MYMEMORY_USER`.

Veja também:

MyMemory, Tradução automática, Sugestões automáticas, MyMemory: API key generator

2.16.82 MT_MYMEMORY_USER

ID de utilizador do MyMemory para a memória de tradução privada. Use-o com `MT_MYMEMORY_KEY`.

Veja também:

MyMemory, Tradução automática, Sugestões automáticas, MyMemory: API key generator

2.16.83 MT_NETEASE_KEY

Chave de app para API da NetEase Sight, pode se registrar em <https://sight.youdao.com/>

Veja também:

NetEase Sight API machine translation, Tradução automática, Sugestões automáticas

2.16.84 MT_NETEASE_SECRET

Segredo de app para a API da NetEase Sight, pode se registrar em <https://sight.youdao.com/>

Veja também:

NetEase Sight API machine translation, Tradução automática, Sugestões automáticas

2.16.85 MT_TMSERVER

URL onde o tmserver está funcionando.

Veja também:

tmserver, Tradução automática, Sugestões automáticas, tmserver

2.16.86 MT_YANDEX_KEY

Chave de API para a API do Yandex Translate, pode registrar-se em <https://yandex.com/dev/translate/>

Veja também:

Yandex Translate, Tradução automática, Sugestões automáticas

2.16.87 MT_YOUDAO_ID

ID do cliente para a API do Youdao Zhiyun, pode registrar-se em <https://ai.youdao.com/product-fanyi-text.s>.

Veja também:

Youdao Zhiyun API machine translation, Tradução automática, Sugestões automáticas

2.16.88 MT_YOUDAO_SECRET

Segredo do cliente para a API do Youdao Zhiyun, pode registrar-se em <https://ai.youdao.com/product-fanyi-text.s>.

Veja também:

Youdao Zhiyun API machine translation, Tradução automática, Sugestões automáticas

2.16.89 MT_SAP_BASE_URL

URL de API ao serviço SAP Translation Hub.

Veja também:

SAP Translation Hub, Tradução automática, Sugestões automáticas

2.16.90 MT_SAP_SANDBOX_APIKEY

Chave de API para uso de API em caixa de proteção

Veja também:

SAP Translation Hub, Tradução automática, Sugestões automáticas

2.16.91 MT_SAP_USERNAME

O seu nome de utilizador da SAP

Veja também:

SAP Translation Hub, Tradução automática, Sugestões automáticas

2.16.92 MT_SAP_PASSWORD

A sua palavra-passe da SAP

Veja também:

SAP Translation Hub, Tradução automática, Sugestões automáticas

2.16.93 MT_SAP_USE_MT

Se se deve também usar serviços de tradução de máquina, além do banco de dados de termos. Valores possíveis: True ou False

Veja também:

SAP Translation Hub, Tradução automática, Sugestões automáticas

2.16.94 NEARBY_MESSAGES

Quantas cadeia devem ser mostradas em torno da cadeia atualmente traduzida. Este é apenas um valor predefinido, os utilizadores podem ajustar-lo em *Perfil do utilizador*.

2.16.95 DEFAULT_PAGE_LIMIT

Novo na versão 4.7.

Default number of elements to display when pagination is active.

2.16.96 PAGURE_CREDENTIALS

Novo na versão 4.3.2.

Lista para credenciais para servidores de Pagure.

Dica: Use isto no caso de querer que o Weblate interaja com mais deles, para um único ponto final do Pagure com *PAGURE_USERNAME* e *PAGURE_TOKEN*.

```
PAGURE_CREDENTIALS = {
  "pagure.io": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "pagure.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

2.16.97 PAGURE_USERNAME

Novo na versão 4.3.2.

O nome de utilizador no Pagure para enviar merge requests para atualizações de tradução.

Veja também:

PAGURE_CREDENTIALS, *Pagure*

2.16.98 PAGURE_TOKEN

Novo na versão 4.3.2.

Token de acesso pessoal do Pagure usado para fazer chamadas API para atualizações de tradução.

Veja também:

PAGURE_CREDENTIALS, *Pagure*, *Pagure API*

2.16.99 RATELIMIT_ATTEMPTS

Novo na versão 3.2.

A quantidade máxima de tentativas de autenticação antes da limitação da taxa ser aplicada.

A predefinição é 5.

Veja também:

Limitação de taxa, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

2.16.100 RATELIMIT_WINDOW

Novo na versão 3.2.

Por quanto tempo a autenticação é aceita após a limitação da taxa ser aplicada.

Uma quantidade de segundos com a predefinição de 300 (5 minutos).

Veja também:

Limitação de taxa, `RATELIMIT_ATTEMPTS`, `RATELIMIT_LOCKOUT`

2.16.101 RATELIMIT_LOCKOUT

Novo na versão 3.2.

Por quanto tempo a autenticação é bloqueada após a limitação da taxa ser aplicada.

Uma quantidade de segundos com a predefinição de 600 (10 minutos).

Veja também:

Limitação de taxa, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`

2.16.102 REGISTRATION_ALLOW_BACKENDS

Novo na versão 4.1.

A lista de backends de autenticação de onde permite o registo. Isso só limita novos registos, os utilizadores ainda se podem autenticar e adicionar autenticação por todos os backends de autenticação configurados.

É recomendado para manter `REGISTRATION_OPEN` ativado enquanto limita os backends de registo, caso contrário, os utilizadores poderão se registar, mas o Weblate não mostrará ligações para se registar na interface do utilizador.

Exemplo:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Dica: Os nomes de backend correspondem aos nomes usados na URL para autenticação.

Veja também:

`REGISTRATION_OPEN`, *Autenticação*

2.16.103 REGISTRATION_CAPTCHA

Um valor de `True` ou `False` indicando se o registo de contas novas é protegido pelo CAPTCHA. Esta configuração é opcional e uma predifinição de `True` será presumido se não for fornecido.

Se for ativado, um CAPTCHA é adicionado a todas as páginas onde um utilizador digita o endereço de e-mail dele:

- Registo de uma conta nova.
- Recuperação de palavra-passe.
- Adição de uma e-mail a uma conta.
- Formulário de contacto para utilizadores que não estão autenticados.

2.16.104 REGISTRATION_EMAIL_MATCH

Novo na versão 2.17.

Permite filtrar quais endereços de e-mail podem ser registados.

A predifinição é `.*`, que permite que registar qualquer endereço de e-mail.

Pode usá-lo para restringir o registo a um único domínio de e-mail:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.105 REGISTRATION_OPEN

Se o registo de contas novas é atualmente permitido. Esta configuração opcional pode permanecer com a predifinição `True` ou pode ser alterada para `Falsa`.

Esta configuração afeta a autenticação embutida por endereço de e-mail ou através do Python Social Auth (pode listar certos back-ends a usar [REGISTRATION_ALLOW_BACKENDS](#)).

Nota: Se estiver a usar métodos de autenticação de terceiros, como [Autenticação por LDAP](#), ele apenas oculta o formulário de registo, mas novos utilizadores ainda conseguem se autenticar e criar contas.

Veja também:

[REGISTRATION_ALLOW_BACKENDS](#), [REGISTRATION_EMAIL_MATCH](#), [Autenticação](#)

2.16.106 REPOSITORY_ALERT_THRESHOLD

Novo na versão 4.0.2.

Limiar para acionar um alerta para repositórios desatualizados ou aqueles que contêm muitas alterações. A predifinição é 25.

Veja também:

[alerts](#)

2.16.107 REQUIRE_LOGIN

Novo na versão 4.1.

Isso ativa :configuração`URLS_DE_LOGIN_NECESSÁRIOS` e configura o framework REST a requisitar autenticação para todos os pontos finais da API.

Nota: Isto é implementado no *Amostra de configuração*. Para Docker, use *WEBLATE_REQUIRE_LOGIN*.

2.16.108 SENTRY_DSN

Novo na versão 3.9.

DSN do Sentry para usar para *Coletando relatórios de erros*.

Veja também:

[Integração Django para o Sentry](#)

2.16.109 IDADE_REGISTRO_SESSÃO_AUTENTICADO

Novo na versão 4.3.

Configura a expiração da sessão para utilizador autenticados. Isso complementa *IDADE_REGISTRO_SESSÃO* que é utilizado por utilizadores não autenticados.

Veja também:

SESSION_COOKIE_AGE

2.16.110 SIMPLIFY_LANGUAGES

Use códigos de idioma simples para combinações predfinidas de idioma/país. Por exemplo, uma tradução de *fr_FR* usará o código de idioma *fr*. Este é geralmente o comportamento desejado, pois simplifica a lista de idiomas para essas combinações predefinidas.

Desative isto se quiser traduções diferentes para cada variante.

2.16.111 SITE_DOMAIN

Configura o domínio do site. Isso é necessário para produzir ligações absolutas corretas em muitos escopos (por exemplo, ativação de e-mails, notificações ou feeds RSS).

No caso de o Weblate estar a ser executado num porte fora do padrão, inclua-a aqui também.

Exemplos::

```
# 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 configuração deve conter apenas o nome de domínio. Para configurar o protocolo (ativar e aplicar HTTPS), use *ENABLE_HTTPS* e para alterar o URL, use *URL_PREFIX*.

Dica: Num contentor Docker, o domínio do site é configurado através de `WEBLATE_ALLOWED_HOSTS`.

Veja também:

Definir domínio correto do site, Configuração de hosts permitidos, Configurar HTTPS corretamente WE-BLATE_SITE_DOMAIN, ENABLE_HTTPS

2.16.112 SITE_TITLE

Título do site a ser usado para o site e e-mails enviados.

2.16.113 SPECIAL_CHARS

Caracteres adicionais para incluir no teclado visual, *Teclado visual*.

O valor predefinido é:

```
SPECIAL_CHARS = ("\t", "\n", "...")
```

2.16.114 SINGLE_PROJECT

Novo na versão 3.8.

Redireciona os utilizadores diretamente para um projeto ou componente em vez de mostrar o painel. Pode configurá-lo como `True` e, neste caso, só funciona no caso de haver realmente apenas um único projeto no Weblate. Alternativamente, define o projeto e redirecionará incondicionalmente para este projeto.

Alterado na versão 3.11: A configuração agora também aceita um slug de projeto, para forçar a exibição desse único projeto.

Exemplo:

```
SINGLE_PROJECT = "test"
```

2.16.115 STATUS_URL

A URL onde a sua instância de Weblate relata o estado dela.

2.16.116 SUGGESTION_CLEANUP_DAYS

Novo na versão 3.2.1.

Apaga sugestões automaticamente após uma determinada quantidade de dias. A predefinição é `None`, ou seja, sem exclusões.

2.16.117 UPDATE_LANGUAGES

Novo na versão 4.3.2.

Controla se o banco de dados de idiomas deve ser atualizado ao executar a migração do banco de dados e está ativado por padrão. Esta configuração não tem efeito na invocação de `setuplang`.

Veja também:

Definições embutidas de idioma

2.16.118 URL_PREFIX

Esta configuração permite que execute Weblate em algum caminho (caso contrário, depende de ser executado a partir da raiz do servidor web).

Nota: Para usar esta configuração, também precisa configurar o seu servidor para remover este prefixo. Por exemplo, com o WSGI, isso pode ser alcançado definindo `WSGIScriptAlias`.

Dica: O prefixo deve iniciar com um `/`.

Exemplo:

```
URL_PREFIX = "/translations"
```

Nota: Esta configuração não funciona com o servidor embutido do Django, teria que ajustar `urls.py` para conter este prefixo.

2.16.119 VCS_BACKENDS

Configuração de backends VCS disponíveis.

Nota: Weblate tenta usar todos os back-ends suportados para os seus utilizadores.

Dica: Pode limitar escolhas ou adicionar back-ends VCS personalizados a usar isto.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Veja também:

Integração de controlo de versões

2.16.120 VCS_CLONE_DEPTH

Novo na versão 3.10.2.

Configura a profundidade a clonagem de repositórios Weblate deve ter.

Nota: Atualmente, isto só é suportado em [Git](#). Por predefinição, o Weblate faz clones rasos dos repositórios para tornar a clonagem mais rápida e economizar espaço no disco. Dependendo do seu uso (por exemplo, ao usar o personalizado [Extensões](#)), pode aumentar a profundidade ou desligar os clones rasos completamente definindo isso para 0.

Dica: No caso de receber o erro fatal: `protocol error: expected old/new/ref, got 'shallow <hash de commit>'` ao fazer push do Weblate, desative clones rasos completamente configurando:

```
VCS_CLONE_DEPTH = 0
```

2.16.121 WEBLATE_ADDONS

Lista de extensões disponíveis para uso. Para usá-las, elas devem ser ativadas para um determinado componente de tradução. Por predefinição, isto inclui todas as extensões embutidas, ao estender a lista, provavelmente vai manter as existentes ativadas, por exemplo:

```
WEBLATE_ADDONS = (  
    # Built-in addons  
    "weblate.addons.gettext.GenerateMoAddon",  
    "weblate.addons.gettext.UpdateLinguasAddon",  
    "weblate.addons.gettext.UpdateConfigureAddon",  
    "weblate.addons.gettext.MsgmergeAddon",  
    "weblate.addons.gettext.GettextCustomizeAddon",  
    "weblate.addons.gettext.GettextAuthorComments",  
    "weblate.addons.cleanup.CleanupAddon",  
    "weblate.addons.consistency.LangaugeConsistencyAddon",  
    "weblate.addons.discovery.DiscoveryAddon",  
    "weblate.addons.flags.SourceEditAddon",  
    "weblate.addons.flags.TargetEditAddon",  
    "weblate.addons.flags.SameEditAddon",  
    "weblate.addons.flags.BulkEditAddon",  
    "weblate.addons.generate.GenerateFileAddon",  
    "weblate.addons.json.JSONCustomizeAddon",  
    "weblate.addons.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: A remoção da extensão da lista não a desinstala dos componentes. Weblate vai travar nesse caso. Por favor, desinstale a extensão de todos os componentes antes de removê-la desta lista.

Veja também:*Extensões, `DEFAULT_ADDONS`*

2.16.122 WEBLATE_EXPORTERS

Novo na versão 4.2.

Lista de exportadores disponíveis que oferecem descarregar traduções ou glossários em vários formatos de ficheiro.

Veja também:*Formatos de ficheiros suportados*

2.16.123 WEBLATE_FORMATS

Novo na versão 3.0.

Lista de formatos de ficheiro disponíveis para uso.

Nota: A lista predfinida já tem os formatos comuns.

Veja também:*Formatos de ficheiros suportados*

2.16.124 WEBLATE_GPG_IDENTITY

Novo na versão 3.1.

Identidade usada pelo Weblate para assinar os commits Git, por exemplo:

```
WEBLATE_GPG_IDENTITY = "Weblate <weblate@example.com>"
```

O chaveiro GPG do Weblate é pesquisado por uma chave correspondente (home/ .gnupg em `DATA_DIR`). Se não for encontrado, uma chave é gerada. Consulte *Signing Git commits with GnuPG* para mais detalhes.

Veja também:*Signing Git commits with GnuPG*

2.16.125 WEBSITE_REQUIRED

Define se *Sítio da Web do Projeto* deve ser especificado ao criar um projeto. Ativado por padrão, pois se adapta às configurações do servidor público.

2.17 Amostra de configuração

The following example is shipped as `weblate/settings_example.py` with Weblate:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
```

(continues on next page)

(continuação da página anterior)

```
# 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.
#

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",
```

(continues on next page)

(continuação da página anterior)

```

        # Uncomment for MySQL older than 5.7:
        # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
        # Set emoji capable charset for MySQL:
        # "charset": "utf8mb4",
        # Change connection timeout in case you get MySQL gone away error:
        # "connect_timeout": 28800,
    },
}

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Data directory
DATA_DIR = os.path.join(BASE_DIR, "data")

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Tagbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),

```

(continues on next page)

(continuação da página anterior)

```

    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "ไทย"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "中文"),
    ("zh-hant", "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

# 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",
)

```

(continues on next page)

(continuação da página anterior)

```

# Make this unique, and don't share it with anybody.
# You can generate it using weblate/examples/generate-secret-key
SECRET_KEY = ""

_TEMPLATE_LOADERS = [
    "django.template.loaders.filesystem.Loader",
    "django.template.loaders.app_directories.Loader",
]
if not DEBUG:
    _TEMPLATE_LOADERS = [("django.template.loaders.cached.Loader", _TEMPLATE_
↪LOADERS)]
TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
            "loaders": _TEMPLATE_LOADERS,
        },
    },
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

```

(continues on next page)


```

SOCIAL_AUTH_BITBUCKET_KEY = ""
SOCIAL_AUTH_BITBUCKET_SECRET = ""
SOCIAL_AUTH_BITBUCKET_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/#account
↪ "

```

(continues on next page)

(continuação da página anterior)

```

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")
    #     }
    # },
]

# 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"

```

(continues on next page)

```

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # Standard Django modules
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.admin.apps.SimpleAdminConfig",
    "django.contrib.admindocs",
    "django.contrib.sitemaps",
    "django.contrib.humanize",
    # Third party Django modules
    "social_django",
    "crispy_forms",
    "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

```

(continues on next page)

(continuação da página anterior)

```

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"},
        "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",

```

(continues on next page)

```

    # },
  },
  "loggers": {
    "django.request": {
      "handlers": ["mail_admins", DEFAULT_LOG],
      "level": "ERROR",
      "propagate": True,
    },
    "django.server": {
      "handlers": ["django.server"],
      "level": "INFO",
      "propagate": False,
    },
    # Logging database queries
    # "django.db.backends": {
    #     "handlers": [DEFAULT_LOG],
    #     "level": "DEBUG",
    # },
    "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Logging VCS operations
    "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Python Social Auth
    "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Django Authentication Using LDAP
    "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # SAML IdP
    "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
  },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    # "weblate.machinery.apertium.ApertiumAPYTranslation",
    # "weblate.machinery.baidu.BaiduTranslation",
    # "weblate.machinery.deepl.DeepLTranslation",
    # "weblate.machinery.glosbe.GlosbeTranslation",
    # "weblate.machinery.google.GoogleTranslation",
    # "weblate.machinery.googlev3.GoogleV3Translation",
    # "weblate.machinery.libretranslate.LibreTranslateTranslation",
    # "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",
    # "weblate.machinery.netease.NeteaseSightTranslation",
    # "weblate.machinery.tmservice.AmagamaTranslation",
    # "weblate.machinery.tmservice.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

```

(continues on next page)

(continuação da página anterior)

```

# DeepL API key
MT_DEEPL_KEY = None

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_Baidu_ID = None
MT_Baidu_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
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

```

(continues on next page)

```

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
# Sent referrrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",) # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails

```

(continues on next page)

(continuação da página anterior)

```

EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PerlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.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.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",

```

(continues on next page)

(continuação da página anterior)

```

# "weblate.checks.chars.NewLineCountCheck",
# "weblate.checks.markup.BBCodeCheck",
# "weblate.checks.chars.ZeroWidthSpaceCheck",
# "weblate.checks.render.MaxSizeCheck",
# "weblate.checks.markup.XMLValidityCheck",
# "weblate.checks.markup.XMLTagsCheck",
# "weblate.checks.markup.MarkdownRefLinkCheck",
# "weblate.checks.markup.MarkdownLinkCheck",
# "weblate.checks.markup.MarkdownSyntaxCheck",
# "weblate.checks.markup.URLCheck",
# "weblate.checks.markup.SafeHTMLCheck",
# "weblate.checks.placeholders.PlaceholderCheck",
# "weblate.checks.placeholders.RegexCheck",
# "weblate.checks.duplicate.DuplicateCheck",
# "weblate.checks.source.OptionalPluralCheck",
# "weblate.checks.source.EllipsisCheck",
# "weblate.checks.source.MultipleFailingCheck",
# "weblate.checks.source.LongUntranslatedCheck",
# "weblate.checks.format.MultipleUnnamedFormatsCheck",
# "weblate.checks.glossary.GlossaryCheck",
# )

# List of automatic fixups
# AUTOFIX_LIST = (
#     "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
#     "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
#     "weblate.trans.autofixes.chars.RemoveZeroSpace",
#     "weblate.trans.autofixes.chars.RemoveControlChars",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LanguaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",
#     "weblate.addons.generate.GenerateFileAddon",
#     "weblate.addons.generate.PseudolocaleAddon",
#     "weblate.addons.json.JSONCustomizeAddon",
#     "weblate.addons.properties.PropertiesSortAddon",
#     "weblate.addons.git.GitSquashAddon",
#     "weblate.addons.removal.RemoveComments",
#     "weblate.addons.removal.RemoveSuggestions",
#     "weblate.addons.resx.ResxUpdateAddon",
#     "weblate.addons.yaml.YAMLCustomizeAddon",
#     "weblate.addons.cdn.CDNJSAddon",
# )

# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

```

(continues on next page)

(continuação da página anterior)

```

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

# List of URLs your site is supposed to serve
ALLOWED_HOSTS = ["*"]

# Configuration for caching
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # "LOCATION": "unix:///var/run/redis/redis.sock?db=1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
            # If you set password here, adjust CELERY_BROKER_URL as well
            "PASSWORD": None,
            "CONNECTION_POOL_KWARGS": {},
        },
        "KEY_PREFIX": "weblate",
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
}

# Store sessions in cache
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# Store messages in session
MESSAGE_STORAGE = "django.contrib.messages.storage.session.SessionStorage"

# REST framework settings for API
REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth` permissions,
    # or allow read-only access for unauthenticated users.
    "DEFAULT_PERMISSION_CLASSES": [
        # Require authentication for login required sites
        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "rest_framework.authentication.SessionAuthentication",
    ),
    "DEFAULT_THROTTLE_CLASSES": (
        "weblate.api.throttling.UserRateThrottle",
        "weblate.api.throttling.AnonRateThrottle",
    ),
    "DEFAULT_THROTTLE_RATES": {"anon": "100/day", "user": "5000/hour"},
    "DEFAULT_PAGINATION_CLASS": ("rest_framework.pagination.PageNumberPagination"),
    "PAGE_SIZE": 20,
    "VIEW_DESCRIPTION_FUNCTION": "weblate.api.views.get_view_description",
}

```

(continues on next page)

```

    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL
FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = [
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": ↪
    ↪ True},
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI": ↪
    ↪ False},
]

# Require login for all URLs
if REQUIRE_LOGIN:
    LOGIN_REQUIRED_URLS = (r"/(.*)$",)

# In such case you will want to include some of the exceptions
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
#     rf"{URL_PREFIX}/accounts/(.*)$", # Required for login
#     rf"{URL_PREFIX}/admin/login/(.*)$", # Required for admin login
#     rf"{URL_PREFIX}/static/(.*)$", # Required for development mode
#     rf"{URL_PREFIX}/widgets/(.*)$", # Allowing public access to widgets
#     rf"{URL_PREFIX}/data/(.*)$", # Allowing public access to data exports
#     rf"{URL_PREFIX}/hooks/(.*)$", # Allowing public access to notification hooks
#     rf"{URL_PREFIX}/healthz/$", # Allowing public access to health check
#     rf"{URL_PREFIX}/api/(.*)$", # Allowing access to API
#     rf"{URL_PREFIX}/js/i18n/$", # JavaScript localization
#     rf"{URL_PREFIX}/contact/$", # Optional for contact form
#     rf"{URL_PREFIX}/legal/(.*)$", # Optional for legal app
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True
# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULE_FILENAME = os.path.join(DATA_DIR, "celery", "beat-schedule")
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate*": {"queue": "translate"},
    "weblate.accounts.tasks.notify*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
}

```

(continues on next page)

(continuação da página anterior)

```

    "weblate.memory.tasks.*": {"queue": "memory"},
}

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

# PGP commits signing
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None
SENTRY_DSN = None
SENTRY_ENVIRONMENT = SITE_DOMAIN
AKISMET_API_KEY = None

```

2.18 Comandos de gerência

Nota: Running management commands under a different user than the one running your webserver can result in files getting wrong permissions, please check *Permissões do sistema de arquivos* 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 Invocando comandos de gestão

As mentioned before, invocation depends on how you installed Weblate.

If using virtualenv for Weblate, you can either specify the full path to **weblate**, or activate the virtualenv prior to invoking it:

```

# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate

```

If you are using source code directly (either from a tarball or Git checkout), the management script is `./manage.py` available in the Weblate sources. To run it:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip 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
```

Veja também:

Instalando a usar Docker, Instalando no Debian e Ubuntu, Instalando no SUSE e openSUSE, Instalando no RedHat, Fedora e CentOS, Instalando a partir do código-fonte

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Novo na versão 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).

Exemplo:

```
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>

Novo na versão 2.5.

Alterado na versão 4.6: Adicionado parâmetro para o modo de tradução.

Performs automatic translation based on other component translations.

--source PROJECT/COMPONENT

Specifies the component to use as source available for translation. If not specified all components in the project are used.

--user USERNAME

Specify username listed as author of the translations. «Anonymous user» is used if not specified.

--overwrite

Whether to overwrite existing translations.

--inconsistent

Whether to overwrite existing translations that are inconsistent (see *Inconsistente*).

--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.

Exemplo:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application_
↩weblate website cs
```

Veja também:

Tradução automática

2.18.4 celery_queues

weblate celery_queues

Novo na versão 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 *Configuração de componente* 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 *Configuração de componente*.

Veja também:

Executar tarefas de manutenção, 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.

Veja também:

Executar tarefas de manutenção

2.18.9 createadmin

weblate createadmin

Creates an `admin` account with a random password, unless it is specified.

--password PASSWORD

Provides a password on the command-line, to not generate a random one.

--no-password

Do not set password, this can be useful with `--update`.

--username USERNAME

Use the given name instead of `admin`.

--email USER@EXAMPLE.COM

Specify the admin e-mail address.

--name

Specify the admin name (visible).

--update

Update the existing user (you can use this to change passwords).

Alterado na versão 2.9: Added parameters `--username`, `--email`, `--name` and `--update`.

2.18.10 dump_memory

weblate dump_memory

Novo na versão 2.20.

Export a JSON file containing Weblate Translation Memory content.

Veja também:

Memória de Tradução, Esquema de memória de tradução Weblate

2.18.11 dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by `importuserdata`

Dica: This comes in handy when migrating or merging Weblate instances.

2.18.12 import_demo

weblate import_demo

Novo na versão 4.1.

Creates a demo project with components based on <<https://github.com/WeblateOrg/demo>>.

This can be useful when developing Weblate.

2.18.13 import_json

weblate import_json <json-file>

Novo na versão 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.

Alterado na versão 2.9: The parameters `--ignore` and `--update` are there to deal with already imported components.

Example of JSON file:

```
[
  {
    "slug": "po",
    "name": "Gettext PO",
    "file_format": "po",
    "filemask": "po/*.po",
    "new_lang": "none"
  },
  {
    "name": "Android",
    "filemask": "android/values-*/strings.xml",
    "template": "android/values/strings.xml",
    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]
```

Veja também:

import_memory

2.18.14 import_memory

weblate import_memory <file>

Novo na versão 2.20.

Imports a TMX or JSON file into the Weblate translation memory.

--language-map LANGMAP

Allows mapping languages in the TMX to the Weblate translation memory. The language codes are mapped after normalization usually done by Weblate.

--language-map en_US:en will for example import all en_US strings as en ones.

This can be useful in case your TMX file locales happen not to match what you use in Weblate.

Veja também:

Memória de Tradução, Esquema de memória de tradução Weblate

2.18.15 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

Alterado na versão 3.0: The import_project command is now based on the *Descoberta de componentes* add-on, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on filemask.

<project> names an existing project, into which the components are to be imported.

The <gitrepo> defines the Git repository URL to use, and <branch> signifies the Git branch. To import additional translation components from an existing Weblate component, use a *weblate://<project>/<component>* URL for the <gitrepo>.

The <filemask> defines file discovery for the repository. It can be either be made simple using wildcards, or it can use the full power of regular expressions.

The simple matching uses **** for component name and *** for language, for example: ***/*.po*

The regular expression has to contain groups named *component* and *language*. For example: *(?P<language>[^\s/]*)(?P<component>[^\s/]*)\.po*

The import matches existing components based on files and adds the ones that do not exist. It does not change already existing ones.

--name-template TEMPLATE

Customize the name of a component using Django template syntax.

For example: *Documentation: {{ component }}*

--base-file-template TEMPLATE

Customize the base file for monolingual translations.

For example: *{{ component }}/res/values/string.xml*

--new-base-template TEMPLATE

Customize the base file for addition of new translations.

For example: *{{ component }}/ts/en.ts*

--file-format FORMAT

You can also specify the file format to use (see *Formatos de ficheiros suportados*), the default is auto-detection.

--language-regex REGEX

You can specify language filtering (see *Configuração de componente*) 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'
```

Veja também:

More detailed examples can be found in the starting chapter, alternatively you might want to use *import_json*.

2.18.16 importuserdata

weblate importuserdata <file.json>

Imports user data from a file created by *dumpuserdata*

2.18.17 importusers

weblate importusers --check <file.json>

Imports users from JSON dump of the Django auth_users database.

--check

With this option it will just check whether a given file can be imported and report possible conflicts arising from usernames or e-mails.

You can dump users from the existing Django installation using:

```
weblate dumpdata auth.User > users.json
```

2.18.18 install_addon

Novo na versão 3.2.

weblate install_addon --addon ADDON <project|project/component>

Instala uma extensão para um conjunto de componentes.

--addon ADDON

Name of the add-on to install. For example *weblate.gettext.customize*.

--configuration CONFIG

Configuração codificada em JSON de uma extensão.

--update

Atualiza a configuração existente da extensão.

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 a saída gettext* for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --
↪update --all
```

Veja também:

Extensões

2.18.19 list_languages

weblate list_languages <locale>

Lists supported languages in MediaWiki markup - language codes, English names and localized names.

This is used to generate <https://wiki.110n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>.

2.18.20 list_translators

weblate list_translators <project|project/component>

Lists translators by contributed language for the given project:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code

List names by language code instead of language name.

You can either define which project or component to use (for example `weblate/application`), or use `--all` to list translators from all existing components.

2.18.21 list_versions

weblate list_versions

Lists all Weblate dependencies and their versions.

2.18.22 loadpo

weblate loadpo <project|project/component>

Reloads translations from disk (for example in case you have done some updates in the VCS repository).

--force

Force update, even if the files should be up-to-date.

--lang LANGUAGE

Limit processing to a single language.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Nota: You seldom need to invoke this, Weblate will automatically load changed files for every VCS update. This is needed in case you manually changed an underlying Weblate VCS repository or in some special cases following an upgrade.

2.18.23 lock_translation

weblate lock_translation <project|project/component>

Prevents further translation of a component.

Dica: 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.

Veja também:

`unlock_translation`

2.18.24 move_language

weblate move_language source target

Novo na versão 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.

Exemplo:

```
weblate move_language cze cs
```

After moving the content, you should check whether there is anything left (this is subject to race conditions when somebody updates the repository meanwhile) and remove the (*generated*) language.

2.18.25 pushgit

weblate pushgit <project|project/component>

Pushes committed changes to the upstream VCS repository.

--force-commit

Force commits any pending changes, prior to pushing.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Nota: Weblate pushes changes automatically if *Enviar ao submeter* in *Configuração de componente* is turned on, which is the default.

2.18.26 unlock_translation

weblate unlock_translation <project|project/component>

Unlocks a given component, making it available for translation.

Dica: 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.

Veja também:

lock_translation

2.18.27 setupgroups

weblate setupgroups

Configures default groups and optionally assigns all users to that default group.

--no-privs-update

Turns off automatic updating of existing groups (only adds new ones).

--no-projects-update

Prevents automatic updates of groups for existing projects. This allows adding newly added groups to existing projects, see *Controlo de acesso ao projeto*.

Veja também:

Lista de privilégios

2.18.28 setuplang

weblate setuplang

Updates list of defined languages in Weblate.

--no-update

Turns off automatic updates of existing languages (only adds new ones).

2.18.29 updatechecks

weblate updatechecks <project|project/component>

Updates all checks for all strings.

Dica: Useful for upgrades which do major changes to checks.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

2.18.30 updategit

weblate updategit <project|project/component>

Fetches remote VCS repositories and updates the internal cache.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components.

Nota: Usually it is better to configure hooks in the repository to trigger *Hooks de notificação*, instead of regular polling by *updategit*.

2.19 Anúncios

Alterado na versão 4.0: Em versões anteriores, esse recurso era chamado de mensagens de quadro de comunicações.

Forneça informações aos seus tradutores postando anúncios, em todo o site, por projeto, componente ou idioma.

Anuncie o propósito, prazos, estados ou especificar metas para tradução.

Os utilizadores receberão notificação sobre os anúncios de projetos assistidos (a menos que optem por não participar).

Isto pode ser útil para várias coisas, desde anunciar o propósito do site até especificar alvos para traduções.

Os anúncios podem ser publicados em cada nível no menu *Manage*, a usar `:guilabel:"Publicar anúncio"`:

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 4.7

[About Weblate](#)

[Legal](#)

[Contact](#)

[Documentation](#)

[Donate to Weblate](#)

Ele também pode ser adicionado a usar a interface administrativa:

2.19. Anúncios

357

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

Os anúncios são então mostrados com base no seu contexto específico:

Nenhum contexto especificado

Mostrado no painel (página de chegada).

Projeto especificado

Mostrado dentro do projeto, incluindo todos os seus componentes e traduções.

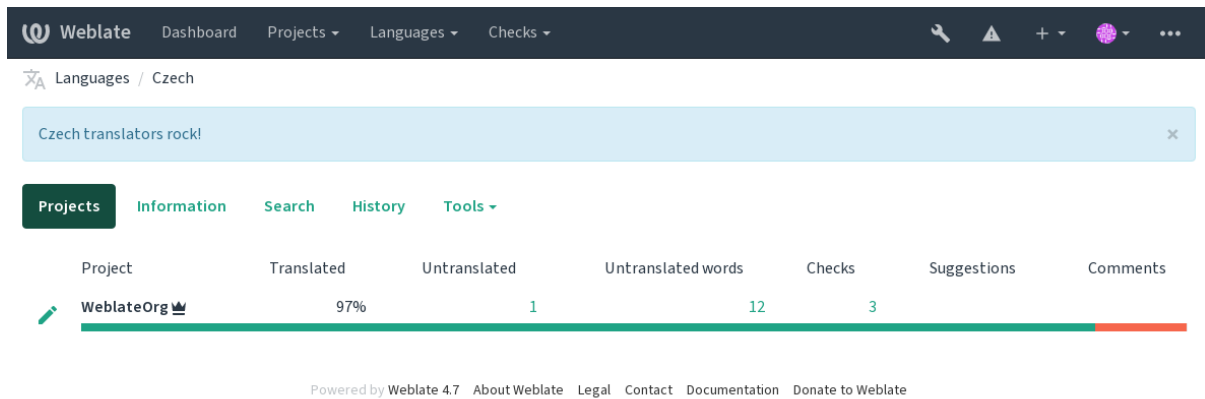
Componente especificado

Mostrado para um determinado componente e todas as traduções dele.

Idioma especificado

Mostrado na visão geral do idioma e todas as traduções nesse idioma.

Esta é a aparência na página de visão geral do idioma:



2.20 Lista de componentes

Especifique múltiplas listas de componentes para aparecer como opções no painel do utilizador, a partir do qual os utilizadores podem seleccionar uma visualização como a visão predefinida. Veja [Painel](#) para saber mais.

Alterado na versão 2.20: Um estado vai ser apresentado para cada componente listado no painel.

Os nomes e conteúdos das listas de componentes podem ser especificados na interface administrativa, na secção *Component lists*. Cada lista de componentes deve ter um nome que é exibido ao utilizador e uma slug representando-a na URL.

Alterado na versão 2.13: Altera as configurações de painel para utilizadores anónimos da interface administrativa, a alterar qual painel é apresentado para utilizadores não autenticados.

2.20.1 Listas de componentes automáticas

Novo na versão 2.13.

Adicione componentes à lista automaticamente com base nas suas slugs criando regras *Automatic component list assignment*.

- Útil para atualizar listas de componentes para grandes instalações, ou no caso de querer ter uma lista de componentes com todos os componentes na sua instalação do Weblate.

Dica: Faça uma lista de componentes contendo todos os componentes da sua instalação 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/WebblateOrg

Chosen components ⓘ

Choose all ⓘ Remove all ⓘ

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE? ⓘ
<input type="text" value="^.*\$"/>	<input type="text" value="^.*\$"/>	<input type="button" value="✕"/>

[+ Add another Automatic component list assignment](#)

2.21 Módulos opcionais do Weblate

Several optional modules are available for your setup.

2.21.1 Exportador git

Novo na versão 2.10.

Provides you read-only access to the underlying Git repository using HTTP(S).

Instalação

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 *Configuração de componente*. 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/'
```

Dica: 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 Faturação

Novo na versão 2.4.

This is used on [Hosted Weblate](#) to define billing plans, track invoices and usage limits.

Instalação

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 do utilizador*.

The billing module additionally allows project admins to create new projects and components without being superusers (see *Adicionando projetos e componentes de tradução*). 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 Legal

Novo na versão 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.

Instalação

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

Veja também:

Cache de avatares, `AVATAR_URL_PREFIX`, `ENABLE_AVATARS`

2.21.5 Proteção contra spam

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:

- Sugestões de utilizador não autenticados
- Descrições e ligações de projetos e componentes

Nota: This (among other things) relies on IP address of the client, please see *Executar por trás de um proxy reverso* for properly configuring that.

Veja também:

Executar por trás de um proxy reverso, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Signing Git commits with GnuPG


Novo na versão 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 2788A52B0C15FAE7E90E3261D2BB2527A9AD025, for which the corresponding public key is found below.

```

-----BEGIN PGP PUBLIC KEY BLOCK-----
mQGNB8GDKQb8BDADAU8TZ1aDIQnhQXc3EacxnZKhnt0faxzLFB3khpL1RSoPpbLR2
aWFDCLmJz+E9W7x0vCFFNGMRG0+d8oKt1R8oMyfZUxGuUdARA8jz8ijM0P0oTHq
bORkujUId/1r+FFKMOAT7ay3NPSVT8njeYprVd2yRyUI6bLIXBzRu5nTlyobxLO
5IE3ZKIMQPoeHIKzkDIDSyhdZ1a9Yf1miTDsvAbDQFgr7do4ZRRHRIEp1tMV4iBnq
D8c4POBexlmj80EdB8IzPr2pnzV8xldaX+qnYUhuIJJkmklpd/814rdjbuGGYjFD
tP6a2XD9xmjHkEWq7bxbvfxJYdoomGbjun4hT9oUL2QNNmp8Co9m3jvQ9/856rRy
DcH2IHEwfWEBbbHvF+ORc7Xw9mXLGzqoGb8nDkLv+9GXPBJChwDmg0GHw61zwQa
gWNTaZWo/eoF5EOInJar/UffRuLlg9qd2GtrOS1MRTAmeUIDyVg5bSk4g9jJyLF6
UasM5mwz8N+jkN0AEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWlQnIkUrdBX1rn6Q4yYdK7JSeprQJQUCYMpbBvwlBawULCQGhAGYV
CgkICwIEFgIDAQIeAQIXgAAKCRAdK7JSeprQJdqDADACNvJI9SuEMP88NkpTwSLgE
XY6ab+kSkjdVxUtmqbdYEHG4r1a0vwXsU9ODYUFWMTY368LZ4AftBb97L8MyxMS
QQBKjebW8syWtiNmRDiEGAbXAelhmu1YOsGwag+PGW5FORV+o3vzAwig2bZtict
FOcDfHwC+7NRrVnt01CRO5/e+dZYcAzAODHFRmInhiw+IIsCsiuaiCn5Fc1hAxln

```

Powered by Weblate 4.7
 [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`.

Veja também:

`WEBLATE_GPG_IDENTITY`

2.21.7 Limitação de taxa

Alterado na versão 3.2: The rate limiting now accepts more fine-grained configuration.

Alterado na versão 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:

Nome	Âmbito	Allowed attempts	Rate limit window	Lockout period
Registo	REGISTRATION	5	300	600
Sending message to admins	MESSAGE	5	300	600
Autenticação de palavra-passe ao entrar	LOGIN	5	300	600
Sítewide search	SEARCH	6	60	60
Traduzir	TRANSLATE	30	60	600
Adding to glossary	GLOSSARY	30	60	600
Iniciando a tradução para um novo idioma	LANGUAGE	2	300	600
Creating new project	PROJECT	5	600	600

If a user fails to log in `AUTH_LOCK_ATTEMPTS` times, password authentication will be turned off on the account until having gone through the process of having its password reset.

The settings can be also applied in the Docker container by adding `WEBLATE_` prefix to the setting name, for example `RATELIMIT_ATTEMPTS` becomes `WEBLATE_RATELIMIT_ATTEMPTS`.

The API has separate rate limiting settings, see *Limitação de taxa da API*.

Veja também:

Limitação de taxa, Executar por trás de um proxy reverso, Limitação de taxa da API

2.21.8 Integração com Fedora Messaging

Fedora Messaging is AMQP-based publisher for all changes happening in Weblate. You can hook additional services on changes happening in Weblate using this.

The Fedora Messaging integration is available as a separate Python module `weblate-fedora-messaging`. Please see https://github.com/WeblateOrg/fedora_messaging/ for setup instructions.

2.22 Personalizar o Weblate

Amplie e personalize a usar Django e Python. Contribua as suas alterações para o upstream acima para que todos possam se beneficiar. Isso reduz os seus custos de manutenção; código no Weblate é cuidado ao alterar interfaces internas ou refatorar o código.

Aviso: Nem interfaces internas nem modelos são considerados uma API estável. Por favor, revise as suas próprias personalizações para cada atualização, as interfaces ou a semântica deles podem mudar sem aviso prévio.

Veja também:

Contribuir ao Weblate

2.22.1 Criar um módulo Python

Se não conhece o Python, pode olhar para [Python For Beginners](#), que explica o básico e aponta aos tutoriais adicionais.

Para escrever algum código Python personalizado (chamado de módulo), é necessário um lugar para armazená-lo, seja no caminho do sistema (geralmente algo como `/usr/lib/python3.7/site-packages/`) ou no diretório Weblate, que também é adicionado ao caminho de pesquisa do interpretador.

Melhor ainda, transforme a sua personalização num pacote Python adequado:

1. Crie uma pasta para o seu pacote (usaremos `weblate_customization`).
2. Dentro dele, crie um ficheiro `setup.py` para descrever o pacote:

```
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. Crie uma pasta para o módulo Python (também chamado de `weblate_customization`) para o código de personalização.
4. Dentro dele, crie um ficheiro `__init__.py` para garantir que o Python possa importar o módulo.
5. Este pacote agora pode ser instalado a usar **`pip install -e`**. Mais informações a serem encontradas em “[Editable](#)” [Installs](#).
6. Uma vez instalado, o módulo pode ser usado na configuração Weblate (por exemplo, `weblate_customization.checks.FooCheck`).

Sua estrutura de módulo deve ser assim:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

Pode encontrar um exemplo de personalização do Weblate em [<https://github.com/WeblateOrg/customize-example>](https://github.com/WeblateOrg/customize-example), ele abrange todos os tópicos descritos abaixo.

2.22.2 Alterar o logotipo

1. Crie uma app Django simples a conter os ficheiros estáticos que deseja substituir (veja [Criar um módulo Python](#)).

A marca aparece nos ficheiros seguintes:

`icons/weblate.svg` Logotipo mostrado na barra de navegação.

`logo-*.png` Ícones web dependendo da resolução do ecrã e do navegador web.

`favicon.ico` Ícone web usado por navegadores legados.

`weblate-*.png` Avatares para bots ou utilizadores anónimos. Alguns navegadores web usam-nos como ícones de atalho.

`email-logo.png` Usado em e-mails de notificações.

2. Adicione-o a `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here...
)
```

3. Execute `weblate collectstatic --noinput`, para coletar ficheiros estáticos servidos aos clientes.

Veja também:

[Managing static files](#) (e.g. images, JavaScript, CSS), [Servir ficheiros estáticos](#)

2.22.3 Verificações de qualidade personalizadas, extensões e correções automáticas

Para instalar o seu código para *Correções automáticas personalizadas*, *Escrever próprias verificações* ou *Escrevendo extensões* no Weblate:

1. Ponha os ficheiros no seu módulo Python a conter a personalização ao Weblate (veja *Criar um módulo Python*).
2. Adicione o caminho totalmente qualificado dele à classe Python nas configurações dedicadas (`WEBLATE_ADDONS`, `CHECK_LIST` ou `AUTOFIX_LIST`):

```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Veja também:

Correções automáticas personalizadas, *Escrever próprias verificações*, *Escrevendo extensões*, *Escrevendo scripts para extensões*

2.23 Interface de gestão

A interface de gestão oferece configurações de administração sob a URL `/manage/`. Está disponível para utilizadores que se inscrevem com privilégios administrativos, acessíveis a usar o ícone da chave inglesa no canto superior direito:

The screenshot shows the Weblate management interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a 'Manage' section with various tabs: 'Weblate status', 'Backups', 'Translation memory', 'Performance report', 'SSH keys', 'Alerts', 'Repositories', 'Users', and 'Appearance'. The 'Weblate status' tab is active, showing the 'Weblate support status' section. This section displays the 'Weblate version' as 4.7 with a unique hash, the 'Support status' as 'Community support', and two buttons: 'Purchase support package' and 'Donate to Weblate'. Below this is the 'Activate support package' section, which includes a description of support packages, an 'Activation token' input field, and buttons for 'Activate' and 'Purchase support package'.

Powered by Weblate 4.7 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Ela inclui uma visão geral básica do seu Weblate:

- Estado de suporte, veja *Obter suporte para o Weblate*
- Backups, veja *Fazer backup e mover o Weblate*

- Memória de tradução compartilhada, veja [Memória de Tradução](#)
- Relatório de desempenho para revisar a saúde do Weblate e a duração das filas do Celery
- Gestão de chaves SSH, veja [Repositórios SSH](#)
- Visão geral de alertas para todos os componentes, veja [alerts](#)

2.23.1 A interface administrativa do Django

Aviso: Será removido no futuro, pois o uso dele é desencorajado — a maioria das funcionalidades pode ser geridas diretamente no Weblate.

Aqui pode gerir objetos armazenados no banco de dados, tais como utilizadores, traduções e outras configurações:

Weblate administration

WELCOME **WEBLATE TEST**
RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT

Site administration

REPORTS

Weblate support status

Status of repositories

SSH keys

Performance report

Translation memory

ACCOUNTS

Audit log entries

+ Add

Change

User profiles

+ Add

Change

Verified e-mails

+ Add

Change

AUTH TOKEN

Tokens

+ Add

Change

AUTHENTICATION

Groups

+ Add

Change

Roles

+ Add

Change

Users

+ Add

Change

BILLING

Billing plans

+ Add

Change

Customer billings

+ Add

Change

Invoices

+ Add

Change

FONTS

Font groups

+ Add

Change

Fonts

+ Add

Change

LEGAL

TOS agreements

+ Add

Change

PYTHON SOCIAL AUTH

Associations

+ Add

Change

Nonces

+ Add

Change

User social auths

+ Add

Change

SCREENSHOTS

Screenshots

+ Add

Change

TRANSLATION MEMORY

Translation memory entries

+ Add

Change

WEBLATE CONFIGURATION

Settings

+ Add

Change

WEBLATE LANGUAGES

Languages

+ Add

Change

WEBLATE TRANSLATIONS

Announcements

+ Add

Change

Component lists

+ Add

Change

Components

+ Add

Change

Contributor agreements

+ Add

Change

Projects

+ Add

Change

Recent actions

My actions

None available

Na secção *Relatórios* pode verificar o estado do seu site, ajustá-lo para produção ou gerir chaves SSH usadas para acessar *Acessando repositórios*.

Gerir objetos de banco de dados em qualquer uma das secções abaixo. A mais interessante é provavelmente *Traduções do Weblate*, onde pode gerir projetos traduzíveis, veja *Configuração de projeto* e *Configuração de componente*.

Idiomas do Weblate detém as definições de idiomas, explicado melhor em *Definições de idioma*.

Adicionar um projeto

A adição de um projeto serve como contentor para todos os componentes. Normalmente cria um projeto para um software, ou livro (Veja *Configuração de projeto* para informações sobre parâmetros individuais):

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:

WebplateOrg

Display name

URL slug:

weblateorg

Name used in URLs and filenames.

Project website:

https://weblate.org/

Main website of translated project.

Translation instructions:

https://weblate.org/contribute/

You can use Markdown and mention users by @username.

☒ Set "Language-Team" header

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

☒ Use shared translation memory

Uses the pool of shared translations between projects.

☒ Contribute to shared translation memory

Contributes to the pool of shared translations between projects.

Access control:

Protected

How to restrict access to this project is detailed in the documentation.

☐ Enable reviews

Requires dedicated reviewers to approve translations.

☐ Enable source reviews

Requires dedicated reviewers to approve source strings.

☒ Enable hooks

Whether to allow updating this repository by remote hooks.

Language aliases:

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

Save and add another

Save and continue editing

SAVE

Veja também:

Configuração de projeto

Componentes bilíngues

Uma vez que adicionou um projeto, os componentes de tradução podem ser adicionados-lo. (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

Webiate administration

WELCOME WEBLATE TEST RETURN TO WEBLATE DOCUMENTATION CHANGE PASSWORD LOGIN/OUT

Home

Project Management

Components

ADD COMPONENT

ADD NEW COMPONENT

ADD

Required fields are marked in bold

Component name

Language names

Required name

URL slug

language-names

Name used in URLs and filenames

Project

WebSiteOrg

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/WebSiteOrg/service

URL of a repository you will use (component to share it with other component)

Repository push URL

URL of a push repository pointing to target of a merge

Repository browser

https://github.com/WebSiteOrg/service/blob/master/FILENAME[FILE]

URL to repository browser, use {branch} for branch, {filename} and {file} as filename and file placeholders

Exported repository URL

URL of repository where users can fetch changes from Weblate

Source string logging reporting address

Email address for reports on errors in source strings. Leave empty for no email

Repository branch

Repository branch to import

Push branch

Branch for pushing changes, leave empty to use repository branch

Filename

weblate/translations/FILENAME[FILE]

Path of file to translate relative to repository root, use * instead of language code, for example path/to/translations/FILENAME[FILE]

Intersubject base language file

Filename of translation base file, containing all strings and their source. It is recommended for intersubject translation formats

☒ Edit base file

Whether users will be able to edit the base file for non-translation formats

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/translations/language.txt

Filename of file used for creating new translations. For gettext choose .pot

File format

gettext PO file

☐ Locked

Locked component will not get any translation updates

☒ Show translation suggestions

Other translation updates in other components will cause automatic translation in this one

☒ Turn on suggestions

Whether to show translation suggestions at all

☐ Suggestive voting

Users can only vote for suggestions and can't make direct translations

Autoscript suggestions

0

Automatically accept suggestions with this number of votes, use 0 to turn it off

Translation flags

Additional custom source string flags to influence Weblate behavior

Enforced checks

List of checks which can not be ignored

Translation license

GNU General Public License v3.0 or later

Contributor agreement

User agreement which needs to be approved before a user can translate this component

Adding new translation

Create new language file

How to handle requests for creating new translations

Language code style

Default based on the file format

Customize language code used to generate the filename for translations created by Weblate

☐ Merge strings

Once adding and removing strings through from Weblate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled

Merge style

Release

Define whether Weblate should merge the upstream repository or release changes into it

Control message when translating

Translated using Weblate {0} language_name {1}
Currently translated at {0} state translated_percent {1} of {0} state all {0} strings
Translated language_name {0} component_name {1}
Translated URL {0} url {1}

You can use template language for various info, please consult the documentation for more details.

Control message when adding translation

Added translation using Weblate {0} language_name {1}

You can use template language for various info, please consult the documentation for more details.

Control message when removing translation

Duplicated translation using Weblate {0} language_name {1}

You can use template language for various info, please consult the documentation for more details.

Control message when merging translation

Merge branch {0} component_name {1} via Weblate

You can use template language for various info, please consult the documentation for more details.

Control message when adding a change

Update translation file
Updated by {0} author_name {1} hook in Weblate
Translated language_name {0} component_name {1}
Translated URL {0} url {1}

You can use template language for various info, please consult the documentation for more details.

☒ Push on commit

Whether the repository should be pushed upstream on every commit

Age of changes to commit

24

Hours to wait after which any pending changes will be committed to the VCS

☒ Lock on error

Whether the component should be locked on repository errors

Source language

English

Language code for source strings in all components

Language filter

includeall

Regular expression used to filter translation files after selecting the format

Variants regular expression

Regular expression used to determine variants of a string

Priority

Medium

Components with higher priority are affected first by translations

☐ Restricted component

Restrict access to the component to only those explicitly given permission

Show subprojects

WebSiteOrg

Choose additional projects where this component will be listed. Hold down "Control" or "Command" on a Mac, to select more than one

☐ Show as a glossary

Glossary color

Silver

Render mode

Localisation

Save and edit again

Save and control editing

Save

Veja também:

Configuração de componente, Formatos bilíngues e monolíngues

Componentes monolínguas

Para facilitar a tradução destes, forneça um ficheiro de modelo contendo o mapeamento de IDs de mensagem para respectivo idioma fonte dele (geralmente inglês). (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

Webiate administration

WELCOME WEBLATE FIRST TIME! WEBLATE - DOCUMENTATION - RELEASES - NEWS - BLOG - HELP

Home - Manage Translations - Components - Add Component

Add Component

SECRET KEYS - SECURITY - ABOUT

Required fields are marked with *

Component name

Required name

URL slug

Required

Project

WebiateOrg

Version control system

Git

Source code repository

webiate/webiateorg/language-strings

Repository push URL

Repository browser

Expected repository URL

Source string log reporting address

Repository branch

Repository branch to translate

Push branch

Filemask

appname/translations/strings and

Modeling base language file

appname/translations/strings and

Use base file

Whether users will be able to edit the base file for modeling translations.

Intermediate language file

Template for new translation

File format

Android String Resource

Locked

Locks component and all its project translation updates.

Allow translation propagation

Whether translation updates in other components will cause automatic translation in this one.

Turn on suggestions

Whether to allow translation suggestions at all.

Suggestion voting

Users can only vote for suggestions and can't make direct translations.

Automatic suggestions

Automatically suggest suggestions with this number of votes, use 0 to turn it off.

Translation flags

Enforced checks

Translation license

MIT License

Contributor agreement

Add new translation

Create new language file

Language code style

Default based on the file format

Merge strings

Enables editing and merging strings straight from WebIDE. If your strings are extracted from the source code or managed externally you probably need to keep it disabled.

Merge style

Follow

Comment message when translating

Comment message when adding translation

Comment message when resuming translation

Comment message when merging translation

Comment message when adding a change

Push on commit

Whether new updates should be pushed upstream on every commit.

Age of changes to commit

54

Lock on error

Whether the component should be locked on repository errors.

Source language

English

Language filter

Git

Variable regular expression

Priority

Medium

Restricted component

Restrict access to the component to only those explicitly given permissions.

Share in projects

WebiateOrg

Use as a glossary

Glossary value

Star

Render revision

Local revision

Save and discard

Save and commit edits

Save

Veja também:

Configuração de componente, Formatos bilíngues e monolíngues

2.24 Obter suporte para o Weblate

Weblate é um software livre protegido por copyleft e com apoio comunitário. Os assinantes recebem apoio prioritário sem custo adicional. Pacotes de ajuda pré-pago estão disponíveis para todos. Pode encontrar mais informações sobre as ofertas de apoio atuais em <<https://weblate.org/support/>>.

2.24.1 Integrando o apoio

Novo na versão 3.8.

Os pacotes de apoio adquiridos podem ser integrados opcionalmente à sua [gestão de assinatura](#) do Weblate, de onde encontrará uma ligação a ele. Detalhes básicos da instância sobre a sua instalação também são relatados de volta ao Weblate desta forma.

2.24.2 Dados enviados ao Weblate

- URL onde a sua instância do Weblate está configurada
- Título do seu site
- A versão do Weblate que está a executar
- Contagem de alguns objetos no seu banco de dados Weblate (projetos, componentes, idiomas, cadeias fonte e utilizadores)
- A chave pública SSH da sua instância

Additionally, when *Descubra Weblate* is turned on:

- List of public projects (name, URL and website)

Nenhum outro dado é enviado.

2.24.3 Serviços de integração

- Veja se o seu pacote de apoio ainda é válido
- *Armazenamento de backup provisionado do Weblate*
- *Descubra Weblate*

Dica: Os pacotes de apoio adquiridos já estão ativados no momento da compra e podem ser usados sem integrá-los.

2.24.4 Descubra Weblate

Novo na versão 4.5.2.

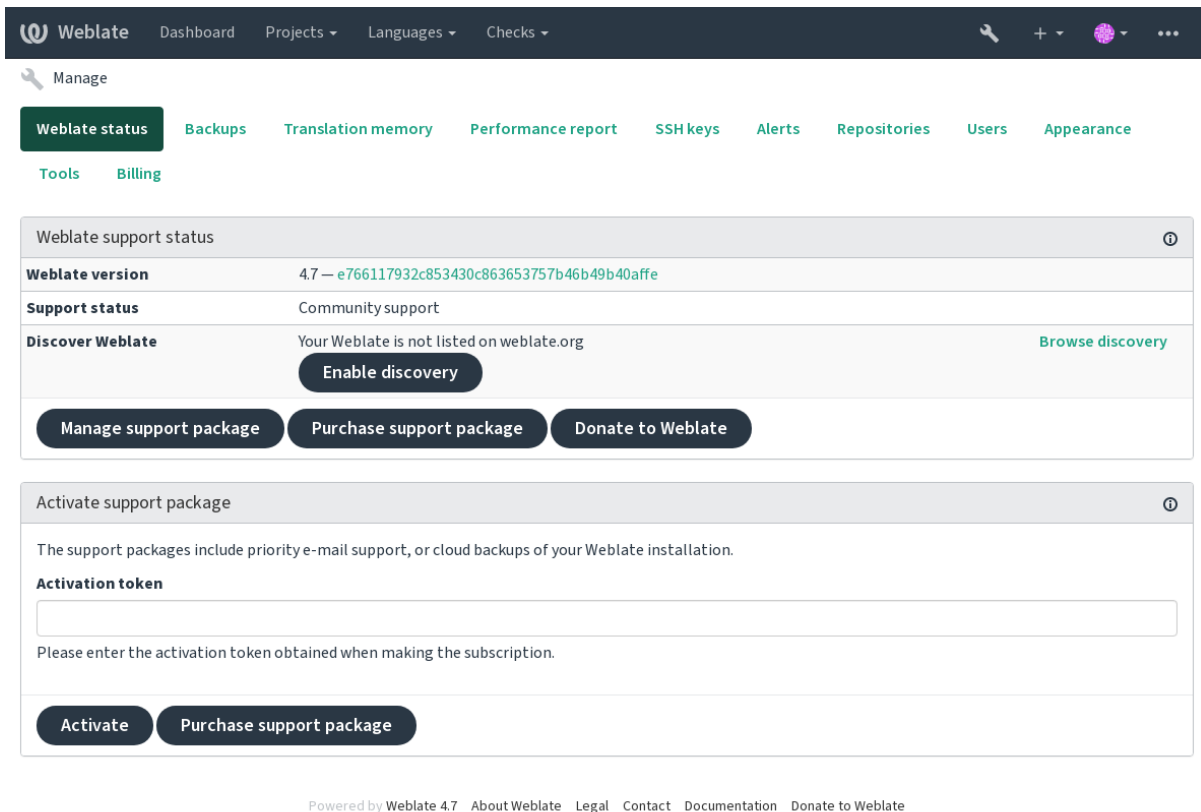
Nota: Este recurso está atualmente num beta inicial.

Descubra Weblate é um serviço opcional que facilita encontrar comunidades e servidores Weblate. Os utilizadores podem navegar por serviços cadastrados em <<https://weblate.org/discover/>> e lá encontrar projetos para contribuir.

Como fazer para ser listado

Dica: Participating in Discover Weblate makes Weblate submit some information about your server, please see *Dados enviados ao Weblate*.

To list your server with an active support subscription (see *Integrando o apoio*) in Discover Weblate all you need to do is turn this on in the management panel:



The screenshot shows the Weblate management interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with a grid of 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 is divided into two panels. The first panel, 'Weblate support status', displays the following information:

- Weblate version:** 4.7 — e766117932c853430c863653757b46b49b40affe
- Support status:** Community support
- Discover Weblate:** Your Weblate is not listed on weblate.org. A button 'Enable discovery' is present, and a link 'Browse discovery' is on the right.

At the bottom of this panel are three buttons: 'Manage support package', 'Purchase support package', and 'Donate to Weblate'.

The second panel, 'Activate support package', contains the following information:

- A description: 'The support packages include priority e-mail support, or cloud backups of your Weblate installation.'
- Activation token:** A text input field.
- A prompt: 'Please enter the activation token obtained when making the subscription.'

At the bottom of this panel are two buttons: 'Activate' and 'Purchase support package'.

At the very bottom of the page, a footer line reads: 'Powered by Weblate 4.7 About Weblate Legal Contact Documentation Donate to Weblate'.

Listing your server without a support subscription in Discover Weblate:

1. Registe-se em <<https://weblate.org/user/>>
2. Registe o seu servidor Weblate no banco de dados de descoberta em <<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:

Webblate status

Webblate version: 4.7 — e766117932c853430c863653757b46b49b40affe

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

Personalizando a listagem

You can customize the listing by providing a text and image (570 x 260 pixels) at <https://weblate.org/user/>.

2.25 Documentos legais

Nota: Aqui encontrará várias informações legais que pode precisar para operar Weblate em certas jurisdições legais. É fornecido como um meio de orientação, sem qualquer garantia de precisão ou correção. Em última análise, é a sua responsabilidade de garantir que o seu uso do Weblate esteja em conformidade com todas as leis e regulamentos aplicáveis.

2.25.1 ITAR e outros controles de exportação

O Weblate pode ser usado dentro do seu próprio datacenter ou nuvem privada virtual. Como tal, ele pode ser usado para armazenar informações ITAR ou outras controladas por exportação; no entanto, os utilizadores finais são responsáveis por garantir tal conformidade.

O serviço Hosted Weblate não foi auditado pela conformidade com ITAR ou outros controles de exportação e atualmente não oferece a capacidade de restringir traduções de acesso por país.

2.25.2 Controlos de criptografia dos EUA

O Weblate não contém nenhum código criptográfico, mas pode ser objeto de controlos de exportação, pois usa componentes de terceiros utilizando criptografia para autenticação, integridade de dados e confidencialidade.

Provavelmente Weblate seria classificado como ECCN 5D002 ou 5D992 e, como software livre publicamente disponível, não deve ser sujeito ao EAR (veja «Itens de criptografia NÃO estão sujeitos a EAR <<https://www.bis.doc.gov/index.php/policy-guidance/encryption/1-encryption-items-not-subject-to-the-ear>>`).

Componentes de software utilizados por Weblate (listando somente os componentes relacionados à função criptográfica):

Python Veja https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG Opcionalmente usado pelo Weblate

Git Opcionalmente usado pelo Weblate

curl Usado pelo Git

OpenSSL Usado pelo Python e cURL

A força de chaves de criptografia depende da configuração do Weblate e os componentes de terceiros que interage com ele, mas em qualquer decente instalação, irá incluir todas as funções criptográficas com exportação restrita:

- Em excesso de 56 bits para um algoritmo simétrico
- Fatorização de inteiros acima de 512 bits para um algoritmo assimétrico
- Cálculo de logaritmos discretos num grupo multiplicativo de um campo finito de tamanho maior do que 512 bits para um algoritmo assimétrico
- Logaritmos discretos num grupo diferente do que acima de 112 bits para um algoritmo assimétrico

O Weblate não tem nenhum recurso de ativação criptográfica, mas pode ser configurado de maneira sem ter nenhum código de criptografia envolvido. Os recursos criptográficos incluem:

- Acessar servidores remotos a usar protocolos seguros (HTTPS)
- Gerar assinaturas para commits de código (PGP)

Veja também:

Controlos de Exportação (EAR) em Software de Código Aberto (*inglês*)

3.1 Contribuir ao Weblate

Há dezenas de maneiras de melhorar o Weblate. Pode escolher uma com a qual se sente confortável, seja codificação, design gráfico, documentação, patrocínio ou uma ideia:

- *Relatando problemas no Weblate*
- *Começar a contribuir código para o Weblate*
- *Traduzir o Weblate*
- *Contribua à documentação do Weblate*
- *Discussões do Weblate*
- *Financiar o desenvolvimento do Weblate*

3.1.1 Traduzir o Weblate

O Weblate está a ser continuamente [traduzido](#) a usar o próprio Weblate. Sinta-se à vontade para participar do esforço de disponibilizar o Weblate na maior quantidade possível de idiomas humanos. Isso traz o Weblate para mais perto dos utilizadores deles!

Se encontrar um possível erro na cadeia fonte, pode marcá-lo com um comentário no editor Weblate. Dessa forma, pode ser discutido e corrigido. Se tiver certeza, também pode clicar na ligação na secção *Localização do texto fonte* e enviar um PR com a sua correção.

3.1.2 Contribua à documentação do Weblate

É bem-vindo para melhorar a página de documentação de sua escolha. Faça isso facilmente a clicar no botão *Editar no GitHub* no canto superior direito da página.

Respeite essas diretrizes ao escrever:

1. Não remova parte da documentação se ela for válida.
2. Use uma linguagem clara e de fácil compreensão. Está a escrever documentos técnicos, não um poema. Nem todos os leitores de documentos são falantes nativos, fique atento.
3. Não tenha medo de perguntar se não tem certeza. Se tiver que perguntar sobre algum recurso durante a edição, não altere os documentos dele antes de ter a resposta. Isso significa: muda ou pergunta. Não faça os dois ao mesmo tempo.
4. Verifique as suas alterações a executar as ações descritas ao seguir os documentos.
5. Envie PR com alterações em pequenos pedaços para tornar mais fácil e rápido revisar e mesclar.
6. Se quiser reescrever e alterar a estrutura de um grande artigo, faça isso em duas etapas:
 1. Reescreva
 2. Depois que a reescrita for revisada, polida e mesclada, altere a estrutura dos parágrafos em outro PR.

Dica: Pode [traduzir os documentos](#).

3.1.3 Discussões do Weblate

Se tiver uma ideia e não tiver certeza se ela é adequada para um problema, não se preocupe. Pode entrar na comunidade de [discussões do GitHub](#).

3.1.4 Financiar o desenvolvimento do Weblate

Pode impulsionar o desenvolvimento do Weblate na [página de doação](#). Os fundos coletados lá são usados para permitir hospedagem de grátis para projetos de software livre e o desenvolvimento adicional do Weblate. Por favor, verifique a [página de doação](#) para obter opções, como metas de financiamento e as recompensas que obtém como um financiador orgulhoso.

Apoiadores que financiaram o Weblate

Lista de apoiadores do Weblate:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- [Cassidy James](#)
- Loic Dachary
- Marozed
- <https://freedombox.org/>
- GNU Solidario (GNU Health)
- [BallotReady](#)
- Richard Nespithal

- [MyExpenses.Mobi](#)

Gostaria de estar na lista? Veja as opções em [Doar ao Weblate](#).

3.2 Começar a contribuir código para o Weblate

Entenda o código-fonte do Weblate a passar por *Código-fonte do Weblate*, *Frontend do Weblate* e *Componentes internos do Weblate*.

3.2.1 Começar com a base de código

Familiarize-se com a base de código do Weblate, a dar uma olhada nos bugs etiquetados como [good first issue](#).

3.2.2 Execução local de Weblate

A abordagem mais confortável para começar a desenvolver Weblate é seguir `.../admin/install/source`. Isso lhe dará um virtualenv com fontes editáveis do Weblate.

1. Clone o código-fonte do Weblate:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Crie um virtualenv:

```
virtualenv .venv
.venv/bin/activate
```

3. Instale Weblate (para isso, vai precisar de algumas dependências do sistema, veja `.../admin/install/source`):

```
pip install -e .
```

3. Instale todas as dependências úteis para o desenvolvimento:

```
pip install -r requirements-dev.txt
```

4. Inicie um servidor de desenvolvimento:

```
weblate runserver
```

5. Dependendo da sua configuração, também quereria inciar workers do Celery:

```
./weblate/examples/celery start
```

6. Para executar um teste (veja *Teste local* para mais detalhes):

```
. scripts/test-database
./manage.py test
```

Veja também:

Instalando a partir do código-fonte

3.2.3 Executando Weblate localmente no Docker

Se tiver Docker e docker-compose instalados, pode pôr para funcionar o ambiente de desenvolvimento simplesmente a executar:

```
./rundev.sh
```

Isso irá criar uma imagem do Docker de desenvolvimento e iniciá-lo. Weblate está a ser executado em `<http://127.0.0.1:8080/>` e pode entrar como o utilizador `admin`, a usar `admin` como a palavra-passe. A nova instalação está vazia, então quererá seguir *Adicionando projetos e componentes de tradução*.

Os ficheiros `Dockerfile` e `docker-compose.yml` para isso estão localizados no diretório `dev-docker`.

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: Tenha cuidado para que os seus contentores Docker estejam prontos e em execução antes de fazer os testes. Pode verificar isso a executar o comando `docker ps`.

Para exibir os logs:

```
./rundev.sh logs
```

Para parar os contentores em segundo fundo, execute:

```
./rundev.sh stop
```

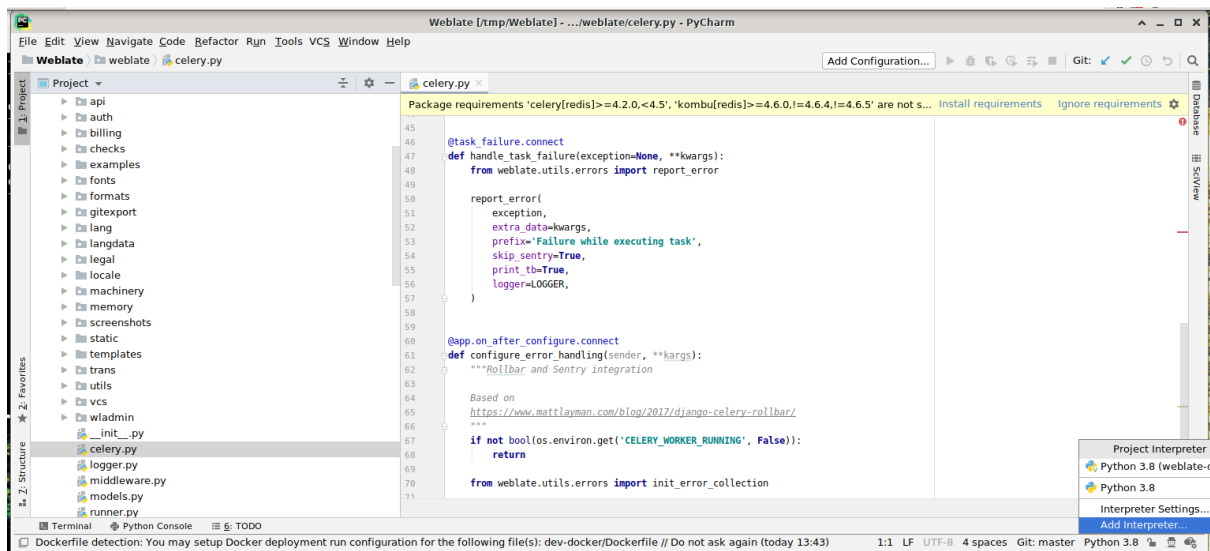
A execução do script sem argumentos vai recriar o contentor Docker e reiniciá-lo.

Nota: Esta não é uma configuração adequada para a produção, pois inclui vários hacks que são inseguros, mas que tornam o desenvolvimento mais fácil.

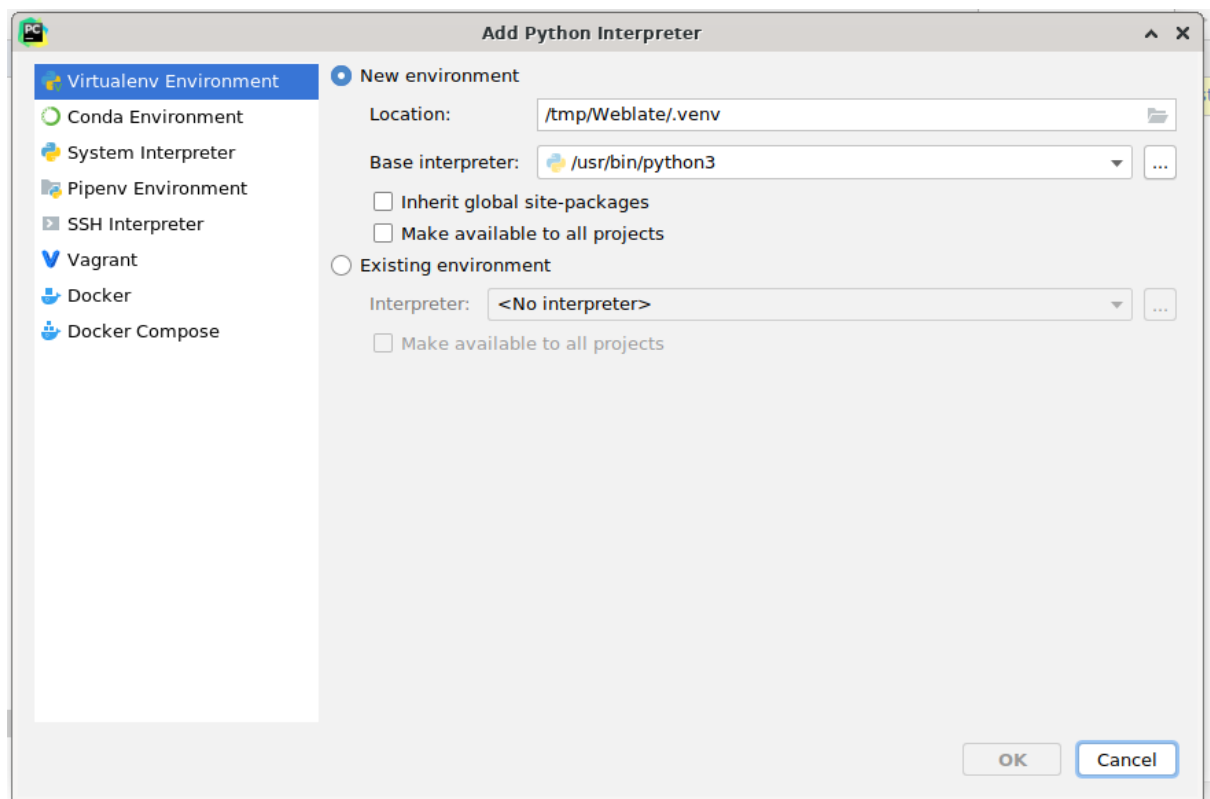
3.2.4 Codificando Weblate com PyCharm

PyCharm é uma IDE conhecida para Python, aqui estão algumas diretrizes para lhe ajudar a configurar o seu projeto Weblate nele.

Considerando que acabou de clonar o repositório GitHub, para uma pasta, basta abri-la com o PyCharm. Assim que a IDE estiver aberta, o primeiro passo é especificar o interpretador que deseja:

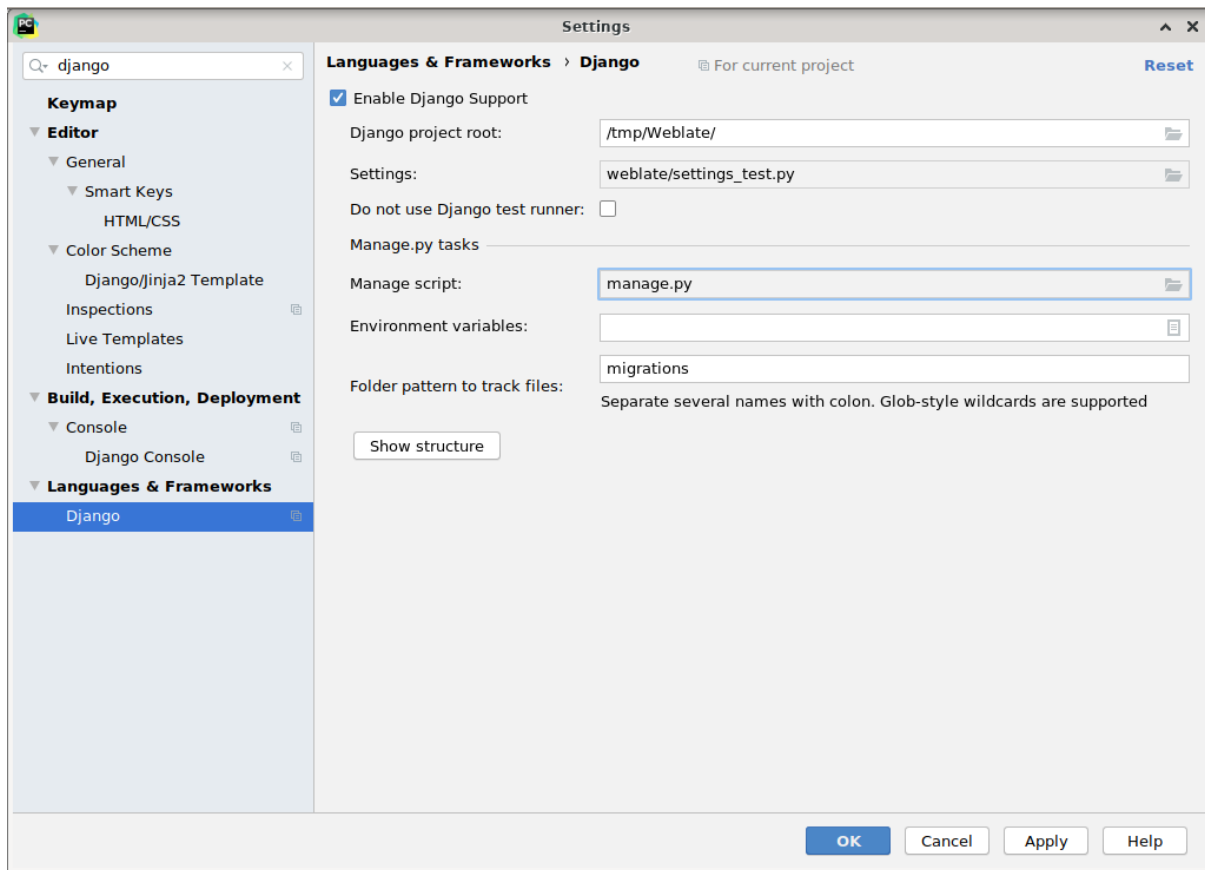


Pode optar por deixar PyCharm criar o virtualenv para si, ou selecionar um já existente:



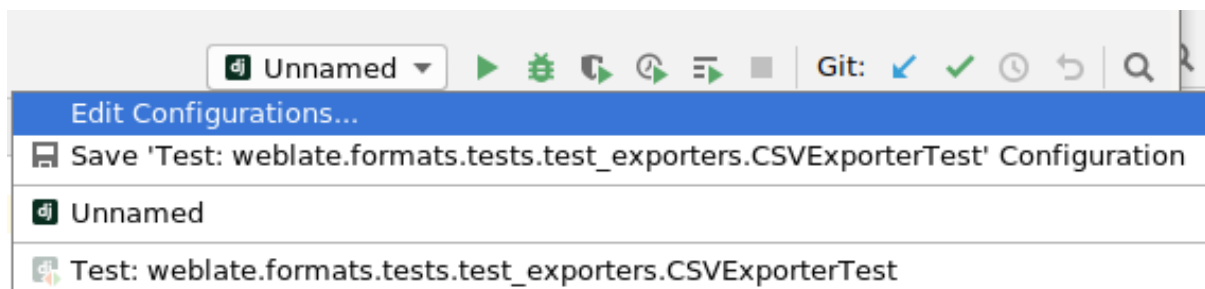
Não se esqueça de instalar as dependências quando o interpretador estiver configurado: faça isso através do console (o console do IDE vai usar diretamente o seu virtualenv por padrão, ou através da interface quando receber um aviso sobre dependências em falta).

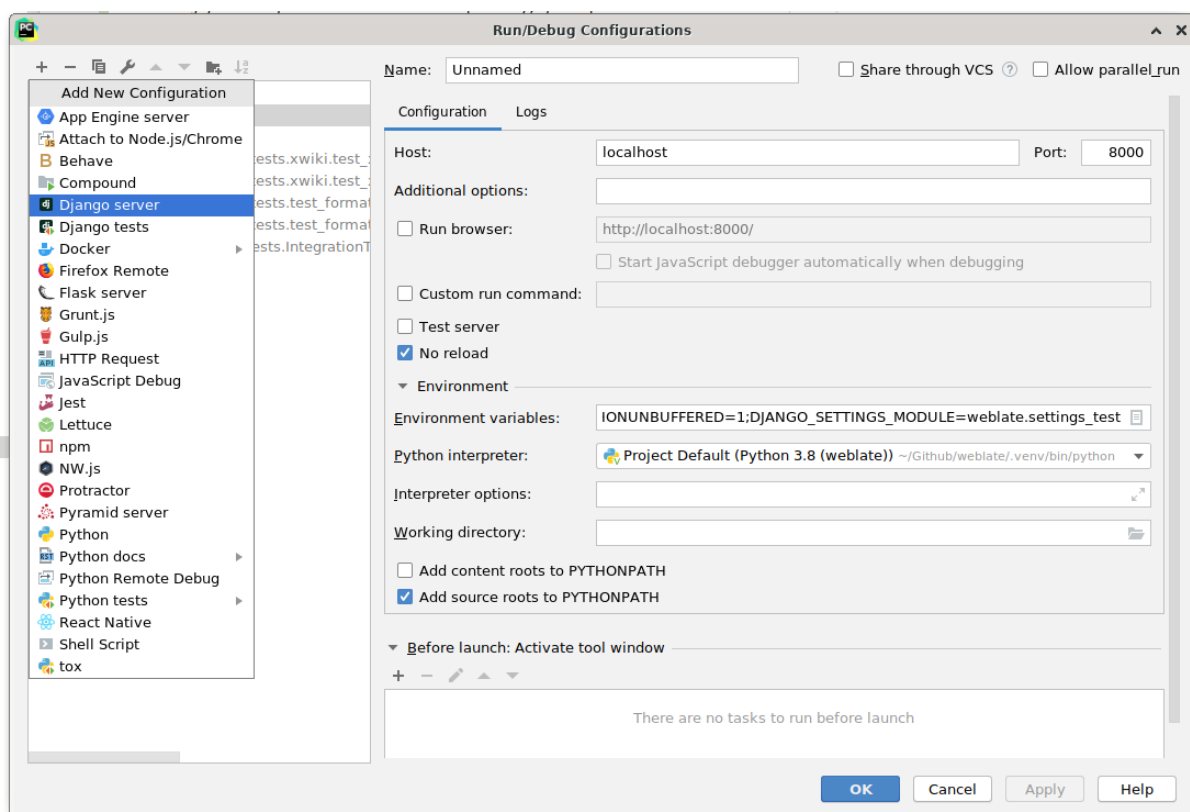
O segundo passo é definir a informação certa para usar o Django nativo dentro do PyCharm: a ideia é ser capaz de acionar imediatamente os testes unitários no IDE. Para isso, precisa especificar o caminho raiz do projeto Django e o caminho para as suas configurações:



Tenha cuidado, *Django project root* é a raiz real do repositório, não o subdiretório do Weblate. Sobre as configurações, poderia usar o `weblate/settings_test.py` do repositório, mas pode criar a sua própria configuração e configurá-la lá.

O último passo é executar o servidor e pôr pontos de interrupção no código para poder depurá-lo. Isto é feito a criar uma nova configuração do *Django Server*:





Dica: Tenha cuidado com a propriedade chamada *No reload*: ela evita do servidor ser recarregado automaticamente quando modificar os ficheiros. Isso permite que os pontos de interrupção de depuração existentes persistam, quando eles normalmente seriam descartados no recarregamento do servidor.

3.2.5 Inicializar a sua instância de desenvolvimento

Quererá usar `import_demo` para criar traduções de demonstração e `createadmin` para criar um utilizador administrador.

3.3 Código-fonte do Weblate

O Weblate é desenvolvido no [GitHub](#). É bem-vindo para criar um fork do código e abrir pull requests. Patches em qualquer outra forma também são bem-vindos.

Veja também:

Confira *Componentes internos do Weblate* para ver como o Weblate se parece por dentro.

3.3.1 Princípios de Segurança por Design

Qualquer código para Weblate deve ser escrito com *Princípios de Segurança por Design* (inglês) em mente.

3.3.2 Padrão de codificação

O código deve seguir as diretrizes de codificação PEP-8 e deve ser formatado a usar o formatador de código **black**.

Para verificar a qualidade do código, pode usar o `:programa:flake8`, os plugins recomendados estão listados em `.pre-commit-config.yaml` e a configuração dele está em `setup.cfg`.

A abordagem mais fácil para impor tudo isso é instalar `pre-commit`. O repositório do Weblate contém a configuração para verificar se os ficheiros do commit estão sãos. Depois de instalá-lo (ele já está incluído no `requirements-lint.txt`), ative-o executando `pré-commit install` na sua cópia do Weblate. Desta forma, todas as suas alterações serão verificadas automaticamente.

Também pode acionar a verificação manualmente, para verificar todos os ficheiros execute:

```
pre-commit run --all
```

3.4 Depuração do Weblate

Os bugs podem se comportar como falhas na aplicação ou como um mau comportamento. É bem-vindo a coletar informações sobre qualquer problema desse tipo e enviá-las para o [rasteador de problemas](#).

3.4.1 Modo de depuração

Ao ligar o modo de depuração, as exceções serão mostradas no navegador. Isto é útil para problemas de depuração na interface web, mas não é adequado para o ambiente de produção, pois tem consequências de desempenho e pode vaziar dados privados.

Veja também:

Desativar o modo de depuração

3.4.2 Logs do Weblate

Weblate pode produzir registos (logs) detalhados do que está a acontecer em segundo fundo. Na configuração padrão, ele usa syslog e isso faz o log aparecer em `/var/log/messages` ou `/var/log/syslog` (a depender da configuração do seu daemon syslog).

O processo de Celery (veja *Tarefas de fundo a usar o Celery*) normalmente produz também registos próprios. O exemplo de configurações de todo o sistema regista em vários ficheiros em `/var/log/celery/`.

Os contentores Docker registam a sua saída (como é habitual no mundo Docker), por isso pode olhar para os logs a utilizar `docker-compose logs`.

Veja também:

Amostra de configuração contém uma configuração de `LOGGING`.

3.4.3 Não a processar tarefas de segundo plano

Muitas coisas acontecem em segundo plano, workers do Celery. Caso coisas como o envio de e-mails ou remoção de componentes não funcionem, pode haver algum problema com isso.

Coisas a verificar neste caso:

- Verifique se o processo do Celery está em execução, veja *Tarefas de fundo a usar o Celery*
- Verifique o estado da fila do Celery em *Interface de gestão* ou use `celery_queues`
- Procure erros nos registos do Celery (veja *Logs do Weblate*)

3.4.4 Não a receber e-mails do Weblate

Pode verificar se o e-mail de saída está a funcionar corretamente a usar o comando de gestão `sendtestemail` (veja *Invocando comandos de gestão* para instruções sobre como invocá-lo em diferentes ambientes) ou a usar *Interface de gestão* sob a guia *Ferramentas*.

Estes enviam e-mails diretamente, para que isto verifique se a sua configuração SMTP está correta (veja *Configuração de e-mail de saída*). A maioria dos e-mails da Weblate são, no entanto, enviados em segundo plano e pode haver alguns problemas com o Celery envolvido também, veja *Não a processar tarefas de segundo plano* para depuração disso.

3.4.5 Análise de falhas de aplicação

Caso o aplicação falhar, é útil recolher o máximo de informação possível sobre a falha. A maneira mais fácil de conseguir isso é a usar serviços de terceiros que podem coletar tais informações automaticamente. Pode encontrar informações sobre como configurar isso em *Coletando relatórios de erros*.

3.4.6 Falhas silenciosas

Muitas tarefas são descarregadas para o Celery para processamento em segundo plano. As falhas não são exibidas na interface do utilizador, mas aparecem nos logs do Celery. A configuração em *Coletando relatórios de erros* ajuda a perceber essas falhas mais facilmente.

3.4.7 Problemas de desempenho

Caso o Weblate tenha um mau desempenho em alguma situação, por favor recolha os registos relevantes a mostrar o problema e qualquer coisa que possa ajudar a descobrir onde o código pode ser melhorado.

Caso alguns pedidos demorem muito tempo sem nenhuma indicação, pode querer instalar `dogslow` juntamente com *Coletando relatórios de erros* e obter tracebacks detalhados na ferramenta de coleta de erros.

3.5 Componentes internos do Weblate

Nota: Este capítulo lhe dará uma visão geral básica dos componentes internos do Weblate.

Weblate deriva a maior parte da estrutura de código dele e é baseado no [Django](#).

3.5.1 Estrutura de diretórios

Visão geral rápida da estrutura de diretórios do repositório principal Weblate:

docs Código-fonte desta documentação, que pode ser construída a usar [Sphinx](#).

dev-docker Código do Docker para executar o servidor de desenvolvimento, veja [Executando Weblate localmente no Docker](#).

weblate Código-fonte do Weblate como uma aplicação [Django](#), veja [Componentes internos do Weblate](#).

weblate/static Ficheiro de cliente (CSS, Javascript e imagens), veja [Frontend do Weblate](#).

3.5.2 Módulos

O Weblate consiste em várias aplicações Django (alguns opcionais, veja [Módulos opcionais do Weblate](#)):

`accounts`

Conta de utilizador, perfis e notificações.

`addons`

Add-ons to tweak Weblate behavior, see [Extensões](#).

`api`

API baseada no [framework Django REST](#).

`auth`

Autenticação e permissões.

`billing`

O módulo opcional de [Faturação](#).

`checks`

Módulo de [Verificações de qualidade](#) de cadeia de tradução.

`fonts`

Módulo de verificações de renderização de fonte.

`formats`

Camada de abstração de formato de ficheiro baseada no `translate-toolkit`.

`gitexport`

O módulo opcional [Exportador git](#).

`lang`

Módulo que define idioma e modelos plurais.

`legal`

O módulo opcional [Legal](#).

`machinery`

Integração de serviços de tradução de máquina.

`memory`

Built-in translation memory, see [Memória de Tradução](#).

`screenshots`

Gestão de capturas de ecrã e módulo OCR.

`trans`

Módulo principal que trata das traduções.

`utils`

Várias utilidades de ajuda.

`vcs`

Abstração do sistema de controle de versão.

`wladmin`

Personalização da interface administrativa do Django.

3.6 Desenvolvendo extensões

Extensões são uma forma de personalizar o fluxo de trabalho de localização no Weblate.

class `weblate.addons.base.BaseAddon` (*storage=None*)

classmethod `can_install` (*component, user*)

Check whether add-on is compatible with given component.

configure (*settings*)

Salva a configuração.

daily (*component*)

Gancho acionado 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*)

Gancho acionado após nova tradução ser adicionada.

post_commit (*component*)

Gancho acionado após alterações terem feito um commit para o repositório.

post_push (*component*)

Gancho acionado após repositório ter o push feito para o upstream.

post_update (*component, previous_head: str, skip_push: bool*)

Gancho acionado após o repositório ser atualizado a partir do upstream.

Parâmetros

- **previous_head** (*str*) – HEAD do repositório antes da atualização, pode estar vazio na clonagem 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*)

Gancho acionado antes das alterações terem feito um commit para o repositório.

pre_push (*component*)

Gancho acionado antes do repositório ter feito push para o upstream.

pre_update (*component*)

Hook acionado antes do repositório ser atualizado a partir do upstream.

save_state ()

Save add-on state information.

stay_on_create = False

Base class for Weblate add-ons.

store_post_load (*translation, store*)

Gancho acionado após um ficheiro ser analisado.

Ele recebe uma instância de uma classe de formato de ficheiro como um argumento.

Isso é útil para modificar os parâmetros de classe de formato de ficheiro, por exemplo, ajustar como o ficheiro será salvo.

unit_pre_create (*unit*)

Gancho acionado antes que a nova unidade seja criada.

Aqui está um exemplo de extensão:

```
#
# Copyright © 2012 - 2021 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <https://www.gnu.org/licenses/>.
#

from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT

class ExampleAddon(BaseAddon):
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {"file_format": {"po", "po-mono"}}
    # List of events add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
    # Detailed add-on description
    description = _("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Frontend do Weblate

O frontend atualmente é construído a usar Bootstrap, jQuery e algumas bibliotecas de terceiros.

3.7.1 Navegadores suportados

Weblate oferece suporte às versões mais recentes e estáveis de todos os principais navegadores e plataformas.

Navegadores alternativos que usam a versão mais recente do WebKit, Blink ou Gecko, seja diretamente ou por meio da API de visualização da web da plataforma, não são explicitamente suportados. No entanto, o Weblate deve (na maioria dos casos) ser exibido e funcionar corretamente nesses navegadores também.

Navegadores mais antigos podem funcionar, mas alguns recursos podem ser limitados.

3.7.2 Gestão de dependências

O gestor de pacotes yarn é usado para atualizar bibliotecas de terceiros. A configuração reside em `scripts/yarn` e há um script wrapper `scripts/yarn-update` para atualizar as bibliotecas, construí-las e copiá-las para os locais corretos em `weblate/static/vendor`, onde todo o terceiro código parcialmente frontend está localizado.

Adicionar uma nova biblioteca de terceiros geralmente consiste em:

```
# 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 Estilo de codificação

Weblate depende do [Prettier](#) para a formatação do código para ficheiros JavaScript e CSS.

Também usamos [ESLint](#) para verificar o código JavaScript.

3.7.4 Tradução

Se precisar de qualquer texto visível para o utilizador no código do frontend, ele deve ser localizável. Na maioria dos casos, tudo que precisa é envolver o seu texto dentro da função `gettext`, mas existem recursos mais complexos disponíveis:

```
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'
```

Veja também:

[Tópico sobre tradução na documentação do Django](#)

3.7.5 Ícones

Weblate atualmente usa ícones de design de material. Caso esteja a procurar por um novo símbolo, verifique [Material Design Icons](#) ou [Material Design Resources](#).

Além disso, existe `scripts/optimize-svg` para reduzir o tamanho do SVG, já que a maioria dos ícones são embutidos no HTML para permitir estilização dos caminhos.

3.8 Relatando problemas no Weblate

O [rastreador de problemas](#) do Weblate está hospedado no GitHub.

Sinta-se à vontade para relatar quaisquer problemas que tenha ou sugerir melhorias para o Weblate lá. Existem vários modelos preparados para lhe orientar confortavelmente durante o relatório de problemas.

Se o que encontrou é um problema de segurança no Weblate, consulte a secção [Problemas de segurança](#) abaixo.

Se não tem certeza sobre o seu relatório de bug ou solicitação de recurso, pode tentar [Discussões do Weblate](#).

3.8.1 Problemas de segurança

A fim de dar à comunidade tempo para responder e fazer upgrade, recomendamos enfaticamente que relate todos os problemas de segurança em particular. O HackerOne é usado para lidar com problemas de segurança e pode ser relatado diretamente em [HackerOne](#). Depois de enviá-lo lá, a comunidade terá tempo limitado, mas suficiente para resolver o incidente.

Como alternativa, relate para security@weblate.org, que termina no HackerOne também.

Se não quiser usar o HackerOne, por qualquer motivo, pode enviar o relatório por e-mail para michal@cihar.com. Pode optar por criptografá-lo a usar esta chave PGP `3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D`. Também pode obter a chave PGP em [Keybase](#).

Nota: O Weblate depende de componentes de terceiros para muitas coisas. Caso encontre uma vulnerabilidade a afetar um desses componentes em geral, relate-o diretamente ao respectivo projeto.

Alguns deles são:

- [Django](#)
 - [framework Django REST](#)
 - [Python Social Auth](#)
-

3.9 Conjunto de testes e integração contínua do Weblate

Existem conjuntos de teste para a maior parte do código atual, aumente a cobertura a adicionar casos de teste para qualquer nova funcionalidade e verifique se funciona.

3.9.1 Integração contínua

Os resultados do teste atual encontram-se em [GitHub Actions](#) e a cobertura é relatada em [Codecov](#).

Existem vários trabalhos para verificar diferentes aspectos:

- Testes de unidade
- Construção de documentação e ligações externas
- Testes de migração de todas as versões com suporte
- Linting de código
- Verificação de configuração (garante que os ficheiros dist gerados não percam nada e possam ser testados)

A configuração do CI está no diretório `.github/workflows`. Ele usa muito scripts auxiliares armazenados no diretório `ci`. Os scripts também podem ser executados manualmente, mas eles requerem várias variáveis de ambiente, principalmente a definir o ficheiro de configurações do Django para usar e conexão com o banco de dados. A definição de exemplo disso está em `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
```

A execução simples pode ser semelhante a:

```
. scripts/test-database
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Teste local

Para executar um conjunto de testes localmente, use:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Dica: Precisar de um servidor de banco de dados (PostgreSQL) a ser usado para os testes. Por padrão, o Django cria um banco de dados separado para executar testes com o prefixo `test_`, então no caso que as suas configurações estarem configuradas para usar `weblate`, os testes usarão o banco de dados `test_weblate`. Veja [Configuração de banco de dados para o Weblate](#) para instruções de configuração.

O `weblate/settings_test.py` é usado no ambiente CI também (veja [Integração contínua](#)) e pode ser ajustado a usar variáveis de ambiente:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}
```

(continues on next page)

(continuação da página anterior)

```
# 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
```

Antes de executar os testes, deve coletar ficheiros estáticos, pois alguns testes contam com a presença deles:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

Também pode especificar testes individuais para executar:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Dica: Os testes também podem ser executados dentro de um contentor Docker do programador, consulte [Executando Weblate localmente no Docker](#).

Veja também:

Veja :doc:`django:topics/testing/index` para mais informações sobre a execução e escrita de testes para Django.

3.10 Esquemas de dados

Weblate usa [JSON Schema](#) para definir o layout de ficheiros JSON externos.

3.10.1 Esquema de memória de tradução Weblate

https://weblate.org/schemas/weblate-memory.schema.json	
tipo	array
items	<i>O item de memória de tradução</i>
tipo	objeto
propriedades	
• categoria	<i>A categoria de cadeia</i>
	1 is global, 2 is shared, 10000000+ are project specific, 20000000+ are user specific
tipo	integer
exemplos	1
minimum	0
predefinido	1
• origem	<i>The String Origin</i>
	Nome do ficheiro ou componente
tipo	cadeia
exemplos	test.tmx
	project/component
predefinido	
• source	<i>A cadeia fonte</i>
tipo	cadeia
exemplos	Olá
minLength	1
predefinido	
• idioma_fonte	<i>O idioma fonte</i>

continues on next page

Table 1 – continuação da página anterior

		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		tipo	cadeia
		exemplos	en
		pattern	^[^]+\$
		predefinido	
	• target	A cadeia de destino	
		tipo	cadeia
		exemplos	Ahoj
		minLength	1
		predefinido	
	• idioma_de_destino	O idioma de destino	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		tipo	cadeia
		exemplos	cs
		pattern	^[^]+\$
		predefinido	
	additionalProperties		False
definições			

Veja também:

Memória de Tradução, dump_memory, import_memory

3.10.2 Exportação de dados de utilizadores do Weblate

https://weblate.org/schemas/weblate-userdata.schema.json			
tipo	objeto		
propriedades			
• basic	Básico		
	tipo	objeto	
	propriedades		
	• nome do uti- lizador	Nome do utilizador	
		tipo	cadeia
		exemplos	administrador
		predefinido	
	• nome_completo	Nome completo	
		tipo	cadeia
		exemplos	Admin Weblate
		predefinido	
	• email	E-mail	
		tipo	cadeia
		exemplos	noreply@example.com
		predefinido	
	• data_de_adeseã	Data de adesão	
		tipo	cadeia
		exemplos	2019-11-18T18:53:54.862Z
		predefinido	
• perfil	Perfil		
	tipo	objeto	
	propriedades		
	• idioma	Idioma	
		tipo	cadeia
		exemplos	cs
		pattern	^.*\$
		predefinido	

continues on next page

Table 2 – continuação da página anterior

	• sugerido	<i>Quantidade de cadeias fonte</i>		
		tipo	<i>integer</i>	
		exemplos	1	
		predefinido	0	
	• traduzido	<i>Quantidade de cadeias traduzidas</i>		
		tipo	<i>integer</i>	
		exemplos	24	
		predefinido	0	
	• enviado	<i>Quantidade de capturas de ecrã enviadas</i>		
		tipo	<i>integer</i>	
		exemplos	1	
		predefinido	0	
	• hide_completed	<i>Ocultar as traduções completas no painel</i>		
		tipo	<i>boolean</i>	
		exemplos	False	
		predefinido	True	
	• secondary_in_zen	<i>Mostrar traduções secundárias no modo Zen</i>		
		tipo	<i>boolean</i>	
		exemplos	True	
		predefinido	True	
	• hide_source_secondary	<i>Ocultar a fonte se existir uma tradução secundária</i>		
		tipo	<i>boolean</i>	
		exemplos	False	
		predefinido	True	
	• hiperligação_do_editor	<i>Hiperligação_do_editor</i>		
		tipo	<i>cadeia</i>	
		exemplos		
		pattern	^.*\$	
	• modo_de_tradução	<i>Modo do editor de tradução</i>		
		tipo	<i>integer</i>	
		exemplos	0	
		predefinido	0	
	• zen_mode	<i>Modo de editor Zen</i>		
		tipo	<i>integer</i>	
		exemplos	0	
		predefinido	0	
	• carateres_especiais	<i>Carateres especiais</i>		
		tipo	<i>cadeia</i>	
		exemplos		
		pattern	^.*\$	
	• vista_do_painel	<i>Vista predefinida do painel</i>		
		tipo	<i>integer</i>	
		exemplos	1	
		predefinido	0	
	• dashboard_component_list	<i>Lista predefinida de componentes</i>		
		predefinido	null	
		anyOf	tipo	<i>null</i>
			tipo	<i>integer</i>
	• idiomas	<i>Idiomas traduzidos</i>		
		tipo	<i>array</i>	
		predefinido		
		items	<i>Código do idioma</i>	
			tipo	<i>cadeia</i>
			exemplos	cs

continues on next page

Table 2 – continuação da página anterior

			pattern	^.*\$			
			predefinido				
	• idiomas_secundários	Idiomas secundários					
		tipo	array				
		predefinido					
		items	Código do idioma				
			tipo	cadeia			
			exemplos	sk			
			pattern	^.*\$			
			predefinido				
	• observado	Projetos vigiados					
		tipo	array				
		predefinido					
		items	Slug do projeto				
			tipo	cadeia			
			exemplos	weblate			
			pattern	^.*\$			
			predefinido				
• registo de auditoria	Registo de auditoria						
	tipo	array					
	predefinido						
	items	Items					
		tipo	objeto				
		propriedades					
		• endereço	Endereço IP				
			tipo	cadeia			
			exemplos	127.0.0.1			
			pattern	^.*\$			
			predefinido				
		• agente_do_utilizador	Agente do utilizador				
			tipo	cadeia			
			exemplos	PC / Linux / Firefox 70.0			
			pattern	^.*\$			
			predefinido				
		• timestamp	Timestamp				
			tipo	cadeia			
exemplos			2019-11-18T18:58:30.845Z				
pattern			^.*\$				
predefinido							
• atividade		Atividade					
		tipo	cadeia				
		exemplos	sessão				
		pattern	^.*\$				
		predefinido					
definições							

Vea também:*Perfil do utilizador, dumpuserdata*

3.11 Criando lançamento do Weblate

3.11.1 Agenda de lançamentos

O Weblate tem um ciclo de lançamento de dois meses para lançamentos (x.y). Geralmente, eles são seguidos por várias versões de correções de bugs para corrigir problemas que ocorrem neles (x.y.z).

A mudança na versão principal indica que o processo de atualização não pode ignorar esta versão - sempre deve atualizar para x.0 antes de atualizar para versões x.y superiores.

Veja também:

Atualizando o Weblate

3.11.2 Planejamento de lançamento

Os recursos para os próximos lançamentos são coletados a usar marcos do GitHub, pode ver nosso roteiro em <<https://github.com/WeblateOrg/weblate/milestones>>.

3.11.3 Processo de lançamento

Coisas para verificar antes de fazer um lançamento:

1. Verificar novos idiomas traduzidos a usar `./scripts/list-translated-languages`.
2. Definir a versão final a usar `./scripts/prepare-release`.
3. Certificar-se de que as capturas de ecrã estejam atualizadas a usar `make -C docs update-screenshots`.
4. Junte todas as traduções pendentes a usar `wlc push; git remote update; git merge origin/weblate`

Realizar o lançamento:

5. Criar um lançamento a usar `./scripts/create-release --tag` (veja abaixo os requisitos).

Etapas manuais pós-lançamento:

6. Atualizar a imagem Docker.
7. Fechar o marco do GitHub.
8. Assim que a imagem Docker for testada, adicionar uma tag e fazer o push dela.
9. Atualizar um chart Helm para a nova versão.
10. Incluir a nova versão em `.github/workflows/migrations.yml` para cobri-la nos testes de migração.
11. Incrementar a versão nas ligações de descarregas do site.
12. Incrementar a versão no repositório a usar `./scripts/set-version`.

Para criar tags a usar o script `./scripts/create-release`, precisará do seguinte:

- GnuPG com chave privada usada para assinar o lançamento
- Acesso por push a repositórios git Weblate (ele envia tags)
- Ferramenta **hub** configurada e acesso para criar lançamentos no repositório Weblate
- Acesso SSH ao servidor de descarregas Weblate (as descarregas do site são copiados lá)

3.12 Segurança e privacidade

Dica: No Weblate, a segurança mantém um ambiente que valoriza a privacidade dos nossos utilizadores.

O desenvolvimento do Weblate segue as [Melhores Práticas da Core Infrastructure Initiative](#) da Linux Foundation.

Veja também:

Problemas de segurança

3.12.1 Rastreamento de dependências para vulnerabilidades

Problemas de segurança em nossas dependências são monitorados a usar [Dependabot](#). Isso cobre as bibliotecas Python e JavaScript e a versão estável mais recente tem as dependências desatualizadas para evitar vulnerabilidades.

Dica: Pode haver vulnerabilidades em bibliotecas de terceiros que não afetam o Weblate, portanto, não são solucionadas com o lançamento de versões de correção de bugs do Weblate.

3.12.2 Segurança de contentor Docker

Os contentores Docker são verificados a usar [Anchore](#) e [Trivy](#).

Isso nos permite detetar vulnerabilidades antecipadamente e lançar melhorias rapidamente.

Pode obter os resultados dessas varreduras no GitHub – eles são armazenados como artefatos em nosso CI no formato SARIF (Static Analysis Results Interchange Format).

Veja também:

Integração contínua

3.13 Sobre o Weblate

3.13.1 Objetivos do projeto

Ferramenta de localização contínua baseada na web com *Integração de controlo de versões* suportando uma ampla gama de *Formatos de ficheiros suportados*, facilitando a contribuição dos tradutores.

3.13.2 Nome do projeto

«Weblate» é uma palavra-valise das palavras «web» e «translate».

3.13.3 Site da Web do Projeto

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>.

3.13.4 Logotipos do projeto

The project logos and other graphics are available in <https://github.com/WeblateOrg/graphics>.

3.13.5 Liderança

This project is maintained by Michal Čihař, who can be reached at michal@cihar.com.

3.13.6 Autores

Weblate was started by Michal Čihař. Since its inception in 2012, thousands of people have contributed.

3.14 Licença

Copyright (C) 2012 - 2021 Michal Čihař <michal@cihar.com>

Este programa é um software livre: pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU, conforme publicado pela Free Software Foundation, seja a versão 3 da Licença, ou (ao seu critério) qualquer versão posterior.

Este programa é distribuído na esperança de que ele seja útil, mas sem qualquer garantia; sem sequer a garantia implícita de COMERCIALIZAÇÃO ou ADEQUAÇÃO PARA UM PROPÓSITO ESPECÍFICO. Consulte a Licença Pública Geral GNU para obter mais detalhes.

Deve ter recebido uma cópia da Licença Pública Geral GNU junto com este programa. Caso contrário, veja <<https://www.gnu.org/licenses/>>.

Histórico de alterações

4.1 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 addon for Windows RC files.
- Fixed possible crash on Xliff updating.

4.2 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.

4.3 Weblate 4.7

Released on June 17th 2021.

- Melhorada a verificação de saúde da configuração.
- Added support for `object-pascal-format` used in gettext PO, see *Formato Object Pascal*.
- Renamed *Nearby keys* to *Similar keys* to better describe the purpose.
- Added support for *mi18n lang files*.
- Improved SAML authentication integration.

- 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.](#)

4.4 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.](#)

4.5 Weblate 4.6.1

Released on May 2nd 2021.

- Remove obsolete spam protection code.
- Melhora na precisão da verificação de plural de textos fonte.
- Update list of user interface languages in Docker.
- Melhoradas as mensagens de erro ao criar pull requests.
- Fixed creating pull requests on Pagure.
- Corrigido acionamento automático de extensões instaladas.
- Corrigidos possíveis problemas de cache ao atualizar.
- Corrigida a adição de novas unidades para traduções monolíngues usando envio.

[Todas as alterações em detalhes.](#)

4.6 Weblate 4.6

Released on April 19th 2021.

- The `auto_translate` management command has now a parameter for specifying translation mode.
- Adicionado suporte para *Ficheiros de texto*.
- Adicionadas tendências e métricas para todos os objetos.
- Adicionado suporte para cópia direta de texto a partir de idiomas secundários.
- Added date filtering when browsing changes.
- Improved activity charts.
- Sender for contact form e-mails can now be configured.
- Melhorada a validação de parâmetros na API de criação de componentes.
- The rate limiting no longer applies to superusers.
- Melhorada a desempenho e confiabilidade da extensão de tradução automática.
- The rate limiting now can be customized in the Docker container.
- API for creating components now automatically uses *URLs internas do Weblate*.
- Simplified state indication while listing strings.
- Hash de palavras-passe agora usa Argon2 por padrão.
- Barras de progresso simplificadas a indicar o estado da tradução.
- Renamed *Adiciona idiomas em falta* to clarify the purpose.
- Corrigido gravar o estado de cadeia para XLIFF.
- Adicionada pesquisa para todo idioma.
- Initial support for *Dimensionando horizontalmente* the Docker deployment.

Todas as alterações em detalhes.

4.7 Weblate 4.5.3

Released on April 1st 2021.

- Fixed metrics collection.
- Corrigido possível travamento ao adicionar textos.
- Melhorados os exemplos de consulta de pesquisa.
- Fixed possible loss of newly added strings on replace upload.

4.8 Weblate 4.5.2

Released on March 26th 2021.

- Agendamento configurável para tradução automática.
- Adicionada verificação de formato Lua.
- Ignore format strings in the *Palavras consecutivas duplicadas* check.
- Allow uploading screenshot from a translate page.

- Added forced file synchronization to the repository maintenance.
- Fixed automatic suggestions for languages with a longer code.
- Melhorado desempenho ao adicionar novos textos.
- Várias correções de erros em verificações de qualidade.
- Several performance improvements.
- Adicionada integração com [Descubra Weblate](#).
- Fixed checks behavior with read-only strings.

[Todas as alterações em detalhes.](#)

4.9 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.
- Store correct source language in TMX files.
- Melhor manuseio para envios de PO monolíngues usando API.
- Improved alerts behavior glossaries.
- Melhoradas verificações de ligações Markdown.
- Indicate glossary and source language in breadcrumbs.
- Listagem paginada de componentes de grandes projetos.
- Aprimorado o desempenho da remoção de tradução, componente ou projeto.
- Improved bulk edit performance.
- Fixed preserving «Needs editing» and «Approved» states for ODF files.
- Interface melhorada para personalização de descarregas de ficheiros de tradução

[Todas as alterações em detalhes.](#)

4.10 Weblate 4.5

Released on February 19th 2021.

- Suporte adicionado para `lua-format` usado em gettext PO.
- Adicionado suporte para compartilhar componente entre projetos.
- Fixed multiple unnamed variables check behavior with multiple format flags.
- Dropped mailing list field on the project in favor of generic instructions for translators.
- Adicionada extensão de geração de pseudolocalidade.
- Suporte adicionado para ficheiros TermBase eXchange.
- Adicionado suporte para definir manualmente variantes de cadeia a usar um marcador.
- Aprimorado o desempenho de verificações de consistência.
- Aprimorado o desempenho de memória de tradução para textos longos.
- Adicionado suporte para pesquisar em explicações.

- Strings can now be added and removed in bilingual formats as well.
- Estende a lista de idiomas suportados na tradução automática do Amazon Translate.
- Ativa automaticamente verificações de MessageFormat para Properties do Java.
- Adicionado um novo método de envio para adicionar novos textos para uma tradução.
- Adicionada uma interface simples para navegação das traduções.
- Glossaries are now stored as regular components.
- Dropped specific API for glossaries as component API is used now.
- Added simplified interface to toggle some of the flags.
- Adicionado suporte para termos não traduzidos ou proibidos no glossário.
- Adicionado suporte para definir terminologia num glossário.
- Moved text direction toggle to get more space for the visual keyboard.
- Adicionada a opção de observar automaticamente os projetos para os quais o utilizador contribuiu.
- Added check whether translation matches the glossary.
- Adicionado suporte para personalizar a cor de texto de navegação.

[Todas as alterações em detalhes.](#)

4.11 Weblate 4.4.2

Released on January 14th 2021.

- Fixed corruption of one distributed MO file.

4.12 Weblate 4.4.1

Released on January 13th 2021.

- Fixed reverting plural changes.
- Fixed displaying help for project settings.
- Administração aprimorada de utilizadores.
- Improved handling of context in monolingual PO files.
- Fixed cleanup add-on behavior with HTML, ODF, IDML and Windows RC formats.
- Fixed parsing of location from CSV files.
- Use content compression for file downloads.
- Improved user experience on importing from ZIP file.
- Improved detection of file format for uploads.
- Avoid duplicate pull requests on Pagure.
- Improved performance when displaying ghost translations.
- Reimplemented translation editor to use native browser textarea.
- Fixed cleanup add-on breaking adding new strings.
- Added API for add-ons.

[Todas as alterações em detalhes.](#)

4.13 Weblate 4.4

Released on December 15th 2020.

- Improved validation when creating a component.
- Weblate now requires Django 3.1.
- Added support for appearance customization in the management interface.
- Fixed read-only state handling in bulk edit.
- Integração com CodeMirror melhorada.
- Added add-on to remove blank strings from translation files.
- The CodeMirror editor is now used for translations.
- Syntax highlighting in translation editor for XML, HTML, Markdown and reStructuredText.
- Highlight placeables in translation editor.
- Improved support for non-standard language codes.
- Added alert when using ambiguous language codes.
- The user is now presented with a filtered list of languages when adding a new translation.
- Extended search capabilities for changes in history.
- Improved billing detail pages and libre hosting workflow.
- Estendida a API de estatísticas de tradução.
- Improved «other translations» tab while translating.
- Added tasks API.
- Improved performance of file upload.
- Improved display of user defined special characters.
- Improved performance of auto-translation.
- Several minor improvements in the user interface.
- Improved naming of ZIP downloads.
- Adicionada a opção para obter notificações de projetos não observados.

[Todas as alterações em detalhes.](#)

4.14 Weblate 4.3.2

Released on November 4th 2020.

- Fixed crash on certain component filemasks.
- Improved accuracy of the consecutive duplicated words check.
- Suporte adicional para solicitações de Pagure.
- Improved error messages for failed registrations.
- Reverted rendering developer comments as Markdown.
- Simplified setup of Git repositories with different default branch than «master».
- Newly created internal repositories now use main as the default branch.
- Reduced false positives rate of unchanged translation while translating reStructuredText.

- Fixed CodeMirror display issues in some situations.
- Renamed Template group to «Sources» to clarify its meaning.
- Fixed GitLab pull requests on repositories with longer paths.

[Todas as alterações em detalhes.](#)

4.15 Weblate 4.3.1

Released on October 21st 2020.

- Melhorada o desempenho da tradução automática.
- Expiração da sessão para utilizadores autenticados corrigida.
- Suporte para ocultar informações da versão adicionado.
- Improve hooks compatibility with Bitbucket Server.
- Improved performance of translation memory updates.
- Reduced memory usage.
- Improved performance of Matrix view.
- Added confirmation before removing a user from a project.

[Todas as alterações em detalhes.](#)

4.16 Weblate 4.3

Released on October 15th 2020.

- Include user stats in the API.
- Fixed component ordering on paginated pages.
- Define source language for a glossary.
- Rewritten support for GitHub and GitLab pull requests.
- Contagens de estatísticas corrigidas após a remoção da sugestão.
- Perfil do utilizador público estendido.
- Fixed configuration of enforced checks.
- Improve documentation about built-in backups.
- Moved source language attribute from project to a component.
- Adicionar a verificação de formatação Vue I18n.
- Generic placeholders check now supports regular expressions.
- Improved look of Matrix mode.
- A maquinaria é agora chamada sugestões automáticas.
- Added support for interacting with multiple GitLab or GitHub instances.
- Extended API to cover project updates, unit updates and removals and glossaries.
- Unit API now properly handles plural strings.
- Component creation can now handle ZIP file or document upload.
- Consolidated API response status codes.

- Support Markdown in contributor agreement.
- Rastreamento de cadeias de origem melhorado.
- Improved JSON, YAML and CSV formats compatibility.
- Suporte adicional para remover cadeias.
- Improved performance of file downloads.
- Improved repository management view.
- Automatically enable java-format for Android.
- Suporte adicional para capturas de ecrã localizadas.
- Suporte adicional para Python 3.9.
- Fixed translating HTML files under certain conditions.

[Todas as alterações em detalhes.](#)

4.17 Weblate 4.2.2

Released on September 2nd 2020.

- Correspondência de cadeias fonte para formatos JSON corrigido.
- Fixed login redirect for some authentication configurations.
- Autenticação LDAP corrigida com sincronização de grupo.
- Falha na comunicação do progresso da tradução automática corrigida.
- Fixed Git commit squashing with trailers enabled.
- Fixed creating local VCS components using API.

4.18 Weblate 4.2.1

Released on August 21st 2020.

- Fixed saving plurals for some locales in Android resources.
- Fixed crash in the cleanup add-on for some XLIFF files.
- Allow setting up localization CDN in Docker image.

4.19 Weblate 4.2

Released on August 18th 2020.

- Improved user pages and added listing of users.
- Dropped support for migrating from 3.x releases, migrate through 4.1 or 4.0.
- Added exports into several monolingual formats.
- Improved activity charts.
- Number of displayed nearby strings can be configured.
- Suporte adicionado para bloquear componentes com erros no repositório.
- Simplified main navigation (replaced buttons with icons).

- Improved language code handling in Google Translate integration.
- The Git squash add-on can generate `Co-authored-by: trailers`.
- Improved query search parser.
- Improved user feedback from format strings checks.
- Improved performance of bulk state changes.
- Added compatibility redirects after project or component renaming.
- Added notifications for strings approval, component locking and license change.
- Added support for ModernMT.
- Allow to avoid overwriting approved translations on file upload.
- Dropped support for some compatibility URL redirects.
- Verificação para literais de modelo de ECMAScript adicionada.
- A opção para observar um componente foi adicionada.
- Removed leading dot from JSON unit keys.
- Fila separada de Celery para memória de tradução foi removida.
- Permitir a tradução de todos os componentes de uma só vez.
- Allow to configure `Content-Security-Policy` HTTP headers.
- Added support for aliasing languages at project level.
- New add-on to help with HTML or JavaScript localization, see *CDN de localização JavaScript*.
- The Weblate domain is now configured in the settings, see *SITE_DOMAIN*.
- Adicionar suporte para a pesquisa por componente e projeto.

4.20 Weblate 4.1.1

Released on June 19th 2020.

- Fixed changing autofix or add-ons configuration in Docker.
- Fixed possible crash in «About» page.
- Improved installation of byte-compiled locale files.
- Fixed adding words to glossary.
- Corrigidos os atalhos de teclado para maquinaria.
- Removed debugging output causing discarding log events in some setups.
- Fixed lock indication on project listing.
- Fixed listing GPG keys in some setups.
- Added option for which DeepL API version to use.
- Added support for acting as SAML Service Provider, see *Autenticação por SAML*.

4.21 Weblate 4.1

Released on June 15th 2020.

- Added support for creating new translations with included country code.
- Added support for searching source strings with screenshot.
- Extended info available in the stats insights.
- Improved search editing on «Translate» pages.
- Improve handling of concurrent repository updates.
- Include source language in project creation form.
- Include changes count in credits.
- Fixed UI language selection in some cases.
- Allow to whitelist registration methods with registrations closed.
- Improved lookup of related terms in glossary.
- Improved translation memory matches.
- Group same machinery results.
- Add direct link to edit screenshot from translate page.
- Improved removal confirmation dialog.
- Include templates in ZIP download.
- Add support for Markdown and notification configuration in announcements.
- Extended details in check listings.
- Added support for new file formats: *Cadeias de PHP Laravel*, *Ficheiro HTML*, *Formato OpenDocument*, *Formato IDML*, *Ficheiros RC do Windows*, *Traduções de INI*, *Traduções Inno Setup INI*, *Propriedades GWT*, *Ficheiros JSON go-i18n*, *Ficheiro ARB*.
- Consistently use dismissed as state of dismissed checks.
- Add support for configuring default add-ons to enable.
- Corrigido o atalho de teclado do editor para dispensar verificações.
- Improved machine translation of strings with placeholders.
- Show ghost translation for user languages to ease starting them.
- Improved language code parsing.
- Show translations in user language first in the list.
- Renamed shapings to more generic name variants.
- Added new quality checks: *Várias variáveis sem nome*, *Não traduzido há muito tempo*, *Palavras consecutivas duplicadas*.
- Reintroduced support for wiping translation memory.
- Fixed option to ignore source checks.
- Added support for configuring different branch for pushing changes.
- API now reports rate limiting status in the HTTP headers.
- Added support for Google Translate V3 API (Advanced).
- Added ability to restrict access on component level.

- Added support for whitespace and other special chars in translation flags, see *Personalizando comportamento a usar marcadores*.
- Always show rendered text check if enabled.
- API now supports filtering of changes.
- Added support for sharing glossaries between projects.

4.22 Weblate 4.0.4

Released on May 7th 2020.

- Fixed testsuite execution on some Python 3.8 environments.
- Typo fixes in the documentation.
- Fixed creating components using API in some cases.
- Fixed JavaScript errors breaking mobile navigation.
- Fixed crash on displaying some checks.
- Fixed screenshots listing.
- Fixed monthly digest notifications.
- Fixed intermediate translation behavior with units non existing in translation.

4.23 Weblate 4.0.3

Released on May 2nd 2020.

- Fixed possible crash in reports.
- User mentions in comments are now case insensitive.
- Fixed PostgreSQL migration for non superusers.
- Fixed changing the repository URL while creating component.
- Fixed crash when upstream repository is gone.

4.24 Weblate 4.0.2

Released on April 27th 2020.

- Improved performance of translation stats.
- Improved performance of changing labels.
- Improved bulk edit performance.
- Melhoria do desempenho da memória de tradução.
- 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.

4.25 Weblate 4.0.1

Released on April 16th 2020.

- Fixed package installation from PyPI.

4.26 Weblate 4.0

Released on April 16th 2020.

- Weblate now requires Python 3.6 or newer.
- Added management overview of component alerts.
- Added component alert for broken repository browser URLs.
- Improved sign in and registration pages.
- Project access control and workflow configuration integrated to project settings.
- Added check and highlighter for i18next interpolation and nesting.
- Added check and highlighter for percent placeholders.
- Mostrar falhas nas verificações de sugestões.
- Record source string changes in history.
- Upgraded Microsoft Translator to version 3 API.
- Reimplemented translation memory backend.
- Added support for several `is :` lookups in *Pesquisando*.
- Allow to make *Tradução inalterada* avoid internal blacklist.
- Improved comments extraction from monolingual po files.
- Renamed whiteboard messages to announcements.
- 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 *Pesquisando*.
- 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.

4.27 Weblate 3.x series

4.27.1 Weblate 3.11.3

Released on March 11th 2020.

- Fixed searching for fields with certain priority.
- Fixed predefined query for recently added strings.
- Fixed searching returning duplicate matches.
- Fixed notifications rendering in Gmail.
- Fixed reverting changes from the history.
- Added links to events in digest notifications.
- Fixed email for account removal confirmation.
- Added support for Slack authentication in Docker container.
- Avoid sending notifications for not subscribed languages.
- Include Celery queues in performance overview.
- Fixed documentation links for add-ons.
- Reduced false negatives for unchanged translation check.
- Raised bleach dependency to address CVE-2020-6802.
- Fixed listing project level changes in history.
- Fixed stats invalidation in some corner cases.
- Fixed searching for certain string states.
- Improved format string checks behavior on missing percent.
- Fixed authentication using some third party providers.

4.27.2 Weblate 3.11.2

Released on February 22nd 2020.

- Fixed rendering of suggestions.
- Fixed some strings wrongly reported as having no words.

4.27.3 Weblate 3.11.1

Released on February 20th 2020.

- Documented Celery setup changes.
- Improved filename validation on component creation.
- Fixed minimal versions of some dependencies.
- Fixed adding groups with certain Django versions.
- Fixed manual pushing to upstream repository.

- Improved glossary matching.

4.27.4 Weblate 3.11

Released on February 17th 2020.

- Allow using VCS push URL during component creation via API.
- Rendered width check now shows image with the render.
- Fixed links in notifications e-mails.
- Improved look of plaintext e-mails.
- Display ignored checks and allow to make them active again.
- Display nearby keys on monolingual translations.
- Suporte adicionado para agrupar formas de cadeias.
- Recommend upgrade to new Weblate versions in the system checks.
- Provide more detailed analysis for duplicate language alert.
- Include more detailed license info on the project pages.
- Automatically unshallow local copies if needed.
- Fixed download of strings needing action.
- New alert to warn about using the same filemask twice.
- Improve XML placeables extraction.
- The `SINGLE_PROJECT` can now enforce redirection to chosen project.
- Added option to resolve comments.
- Added bulk editing of flags.
- Added support for labels.
- Added bulk edit add-on.
- Added option for *Forçar verificações*.
- Increased default validity of confirmation links.
- Improved Matomo integration.
- Fixed *Foi traduzido* to correctly handle source string change.
- Extended automatic updates configuration by `AUTO_UPDATE`.
- LINGUAS add-ons now do full sync of translations in Weblate.

4.27.5 Weblate 3.10.3

Released on January 18th 2020.

- Support for translate-toolkit 2.5.0.

4.27.6 Weblate 3.10.2

Released on January 18th 2020.

- Add lock indication to projects.
- Fixed CSS bug causing flickering in some web browsers.
- Fixed searching on systems with non-English locales.
- Improved repository matching for GitHub and Bitbucket hooks.
- Fixed data migration on some Python 2.7 installations.
- Allow configuration of Git shallow cloning.
- Improved background notification processing.
- Fixed broken form submission when navigating back in web browser.
- New add-on to configure YAML formatting.
- Fixed same plurals check to not fire on single plural form languages.
- Fixed regex search on some fields.

4.27.7 Weblate 3.10.1

Released on January 9th 2020.

- Extended API with translation creation.
- Fixed several corner cases in data migrations.
- Compatibility with Django 3.0.
- Improved data clean-up performance.
- Added support for customizable 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.

4.27.8 Weblate 3.10

Released on December 20th 2019.

- Interface de utilizador da aplicação melhorada.
- Added doublespace check.
- Fixed creating new languages.
- Avoid sending auditlog notifications to deleted e-mails.
- Added support for read only strings.
- Added support for Markdown in comments.
- Allow placing translation instruction text in project info.
- Add copy to clipboard for secondary languages.
- Improved support for Mercurial.
- Improved Git repository fetching performance.

- Add search lookup for age of string.
- Show source language for all translations.
- Show context for nearby strings.
- 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.
- Improved look of notification e-mails.
- Provide choice for translation license.

4.27.9 Weblate 3.9.1

Released on October 28th 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.

4.27.10 Weblate 3.9

Released on October 15th 2019.

- Include Weblate metadata in downloaded files.
- Improved UI for failing checks.
- Indicate missing strings in format checks.
- Separate check for French punctuation spacing.
- Add support for fixing some of quality checks errors.
- Add separate permission to create new projects.
- Extend stats for char counts.
- Improve support for Java style language codes.

- Added new generic check for placeholders.
- Added support for WebExtension JSON placeholders.
- 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.

4.27.11 Weblate 3.8

Released on August 15th 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.
- Improved look of notification e-mails.
- Fixed password reset behavior.
- Improved performance on most of translation pages.
- 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.

4.27.12 Weblate 3.7.1

Released on June 28th 2019.

- Documentation updates.
- Fixed some requirements constraints.
- Updated language database.
- Localization updates.
- Various user interface tweaks.
- Improved handling of unsupported but discovered translation files.
- More verbosely report missing file format requirements.

4.27.13 Weblate 3.7

Released on June 21st 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.

4.27.14 Weblate 3.6.1

Released on April 26th 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.

4.27.15 Weblate 3.6

Released on April 20th 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.
- Added notifications for new alerts, whiteboard messages or components.
- Notifications for administered projects can now be configured.
- Improved handling of three letter language codes.

4.27.16 Weblate 3.5.1

Released on March 10th 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.
- Allow to edit file format in component settings.
- Update installation instructions to prefer Python 3.
- Performance and consistency improvements for loading translations.
- Make Microsoft Terminology service compatible with current Zeep releases.
- Localization updates.

4.27.17 Weblate 3.5

Released on March 3rd 2019.

- Improved performance of built-in translation memory.
- Added interface to manage global translation memory.
- Improved alerting on bad component state.
- Added user interface to manage whiteboard messages.
- 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.
- Improved navigation in the Zen mode.
- Added several new quality checks (Markdown related and URL).

- Added support for app store metadata files.
- Added support for toggling GitHub or Gerrit integration.
- Added check for Kashida letters.
- Added option to squash commits based on authors.
- Improved support for XLSX file format.
- Compatibility with Tesseract 4.0.
- Billing add-on now removes projects for unpaid billings after 45 days.

4.27.18 Weblate 3.4

Released on January 22nd 2019.

- Added support for XLIFF placeholders.
- Celery can now utilize multiple task queues.
- Added support for renaming and moving projects and components.
- Include characters counts in reports.
- Added guided adding of translation components with automatic detection of translation files.
- Customizable merge commit messages for Git.
- Added visual indication of component alerts in navigation.
- Improved performance of loading translation files.
- New add-on to squash commits prior to push.
- Improved displaying of translation changes.
- Changed default merge style to rebase and made that configurable.
- Better handle private use subtags in language code.
- Improved performance of fulltext index updates.
- Extended file upload API to support more parameters.

4.27.19 Weblate 3.3

Released on November 30th 2018.

- Added support for component and project removal.
- Improved performance for some monolingual translations.
- Added translation component alerts to highlight problems with a translation.
- Expose XLIFF string resname as context when available.
- Added support for XLIFF states.
- Added check for non writable files in DATA_DIR.
- Improved CSV export for changes.

4.27.20 Weblate 3.2.2

Released on October 20th 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.

4.27.21 Weblate 3.2.1

Released on October 10th 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.

4.27.22 Weblate 3.2

Released on October 6th 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.

4.27.23 Weblate 3.1.1

Released on July 27th 2018.

- Fix testsuite failure on some setups.

4.27.24 Weblate 3.1

Released on July 27th 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.
- Nova extensão para marcar traduções não alteradas como a necessitar de edição.
- 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.

4.27.25 Weblate 3.0.1

Released on June 10th 2018.

- Fixed possible migration issue from 2.20.
- Localization updates.
- Removed obsolete hook examples.
- Improved caching documentation.
- Fixed displaying of admin documentation.
- Improved handling of long language names.

4.27.26 Weblate 3.0

Released on June 1st 2018.

- Rewritten access control.
- Several code cleanups that lead to moved and renamed modules.
- Nova extensão para descoberta automática de componentes.
- 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.

4.28 Weblate 2.x series

4.28.1 Weblate 2.20

Released on April 4th 2018.

- Improved speed of cloning subversion repositories.
- Changed repository locking to use third party library.
- Added support for downloading only strings needing action.
- Added support for searching in several languages at once.
- New add-on to configure gettext output wrapping.
- 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.
- Added support for HTML markup in whiteboard messages.
- Added support for mass changing state of strings.
- Translate-toolkit at least 2.3.0 is now required, older versions are no longer supported.
- Adicionada memória de tradução embutida.
- 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.
- Adicionado suporte para reordenar alterações de commits feitos.

4.28.2 Weblate 2.19.1

Released on February 20th 2018.

- Fixed migration issue on upgrade from 2.18.
- Improved file upload API validation.

4.28.3 Weblate 2.19

Released on February 15th 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.
- Melhorado o relatório de progresso de tradução a nível de idioma.
- Improved support for different plural formulas.
- Added support for Subversion repositories not using stdlayout.
- Added add-ons to customize translation workflows.

4.28.4 Weblate 2.18

Released on December 15th 2017.

- Extended contributor stats.
- Improved configuration of special characters virtual keyboard.
- Added support for DTD file format.
- Alterados os atalhos do teclado para reduzir a chance de colisão com os do navegador/sistema.
- 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.

4.28.5 Weblate 2.17.1

Released on October 13th 2017.

- Fixed running testsuite in some specific situations.
- Locales updates.

4.28.6 Weblate 2.17

Released on October 13th 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.

4.28.7 Weblate 2.16

Released on August 11th 2017.

- Various performance improvements.
- Added support for nested JSON format.
- Added support for WebExtension JSON format.
- Corrigida a autenticação com exportador git.
- 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.

4.28.8 Weblate 2.15

Released on June 30th 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.

4.28.9 Weblate 2.14.1

Released on May 24th 2017.

- Fixed possible error when paginating search results.
- Fixed migrations from older versions in some corner cases.
- Fixed possible CSRF on project watch and unwatch.
- The password reset no longer authenticates user.
- Fixed possible CAPTCHA bypass on forgotten password.

4.28.10 Weblate 2.14

Released on May 17th 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.

4.28.11 Weblate 2.13.1

Released on Apr 12th 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.

4.28.12 Weblate 2.13

Released on Apr 12th 2017.

- Fixed quality checks on translation templates.
- Added quality check to trigger on losing translation.
- Add option to view pending suggestions from user.
- Add option to automatically build component lists.
- Default dashboard for unauthenticated users can be configured.
- Add option to browse 25 random strings for review.
- History now indicates string change.
- 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.

4.28.13 Weblate 2.12

Released on Mar 3rd 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.

4.28.14 Weblate 2.11

Released on Jan 31st 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.
- Incluído por projecto e estatísticas de componentes.
- 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.

4.28.15 Weblate 2.10.1

Released on Jan 20th 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

4.28.16 Weblate 2.10

Released on Dec 15th 2016.

- Added quality check to check whether plurals are translated differently.
- Fixed GitHub hooks for repositories with authentication.
- Adicionado módulo opcional de exportador git.
- 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.

4.28.17 Weblate 2.9

Released on Nov 4th 2016.

- Extended parameters for createadmin management command.
- Extended import_json to be able to handle with existing components.
- Added support for YAML files.
- Project owners can now configure translation component and project details.
- Use «Watched» instead of «Subscribed» projects.
- Projects can be watched directly from project page.
- Added multi language status widget.
- Highlight secondary language if not showing source.
- Record suggestion deletion in history.
- Improved UX of languages selection in profile.
- Fixed showing whiteboard messages for component.
- Keep preferences tab selected after saving.
- Show source string comment more prominently.
- Automatically install Gettext PO merge driver for Git repositories.
- Added search and replace feature.
- Added support for uploading visual context (screenshots) for translations.

4.28.18 Weblate 2.8

Released on Aug 31st 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.

4.28.19 Weblate 2.7

Released on Jul 10th 2016.

- Removida a tradução automática da web do Google.
- 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.

4.28.20 Weblate 2.6

Released on Apr 28th 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.
- Added categories to whiteboard messages.

4.28.21 Weblate 2.5

Released on Mar 10th 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.
- Performance improvements for quality checks.
- Pesquisa corrigida para todo o site por verificações com falha.

- Added option to specify source language.
- Improved support for XLIFF files.
- Extended list of options for `import_project`.
- Improved targeting for whiteboard messages.
- Support for automatic translation across projects.
- Optimized fulltext search index.
- Added management command for auto translation.
- Added placeables highlighting.
- Adicionados atalhos de teclado para os objetos colocáveis, verificações e traduções automáticas.
- Improved translation locking.
- Added quality check for AngularJS interpolation.
- Added extensive group based ACLs.
- Clarified terminology on strings needing edit (formerly fuzzy).
- Clarified terminology on strings needing action and not translated strings.
- Support for Python 3.
- Dropped support for 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.

4.28.22 Weblate 2.4

Released on Sep 20th 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.
- Estendidos os atalhos de teclado.
- 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.

4.28.23 Weblate 2.3

Released on May 22nd 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.

4.28.24 Weblate 2.2

Released on Feb 19th 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.

4.28.25 Weblate 2.1

Released on Dec 5th 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.
- Various bugfixes.
- 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.

4.28.26 Weblate 2.0

Released on Nov 6th 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.

4.29 Weblate 1.x series

4.29.1 Weblate 1.9

Released on May 6th 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.

4.29.2 Weblate 1.8

Released on November 7th 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.
- Users can login by e-mail instead of username.
- Documentation improvements.
- Improved source strings review.
- Searching across all strings.

- Better tracking of source strings.
- Captcha protection for registration.

4.29.3 Weblate 1.7

Released on October 7th 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.
- Changes in dictionary are now logged in history.
- Performance improvements for translating view.

4.29.4 Weblate 1.6

Released on July 25th 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.

4.29.5 Weblate 1.5

Released on April 16th 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.

4.29.6 Weblate 1.4

Released on January 23rd 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.

4.29.7 Weblate 1.3

Released on November 16th 2012.

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Melhoria do processamento dos controlos de qualidade.
- Added new checks (BB code, XML markup and newlines).
- Support for optional rebasing instead of merge.
- Possibility to relocate Weblate (for example to run it under /weblate path).
- Support for manually choosing file type in case autodetection fails.
- Better support for Android resources.
- Support for generating SSH key from web interface.
- More visible data exports.
- New buttons to enter some special characters.
- Support for exporting dictionary.
- Support for locking down whole Weblate installation.
- Checks for source strings and support for source strings review.
- Support for user comments for both translations and source strings.
- Better changes log tracking.
- Changes can now be monitored using RSS.
- Improved support for RTL languages.

4.29.8 Weblate 1.2

Released on August 14th 2012.

- Weblate now uses South for database migration, please check upgrade instructions if you are upgrading.
- Fixed minor issues with linked git repos.
- New introduction page for engaging people with translating using Weblate.
- 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.
- Fazer cache de contagens de cadeias a conter verificações com falha.
- Checking of requirements during setup.
- Documentation improvements.

4.29.9 Weblate 1.1

Released on July 4th 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.

4.29.10 Weblate 1.0

Released on May 10th 2012.

- Improved validation while adding/saving component.
- Experimental support for Android component files (needs patched ttkit).
- Updates from hooks are run in background.
- Improved installation instructions.
- Improved navigation in dictionary.

4.30 Weblate 0.x series

4.30.1 Weblate 0.9

Released on April 18th 2012.

- Fixed import of unknown languages.
- Improved listing of nearby messages.
- Improved several checks.
- Documentation updates.

- Added definition for several more languages.
- Várias limpezas de código.
- Documentation improvements.
- Alteração do layout do ficheiro.
- Update helper scripts to Django 1.4.
- Improved navigation while translating.
- Better handling of po file renames.
- Better validation while creating component.
- Integrated full setup into syncdb.
- Added list of recent changes to all translation pages.
- Check for not translated strings ignores format string only messages.

4.30.2 Weblate 0.8

Released on April 3rd 2012.

- Replaced own full text search with Whoosh.
- Various fixes and improvements to checks.
- New command updatechecks.
- Lot of translation updates.
- Added dictionary for storing most frequently used terms.
- Added /admin/report/ for overview of repositories status.
- 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.

4.30.3 Weblate 0.7

Released on February 16th 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.

4.30.4 Weblate 0.6

Released on February 14th 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.

4.30.5 Weblate 0.5

Released on February 12th 2012.

- **Support for machine translation using following online services:**
 - Apertium
 - Microsoft Translator
 - MyMemory
- Várias traduções novas.
- 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).

4.30.6 Weblate 0.4

Released on February 8th 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

4.30.7 Weblate 0.3

Released on February 8th 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.

4.30.8 Weblate 0.2

Released on February 7th 2012.

- Improved validation of several forms.
- Warn users on profile upgrade.
- Lembre-se de URL para fazer o login.
- Naming of text areas while entering plural forms.
- Automatic expanding of translation area.

4.30.9 Weblate 0.1

Released on February 6th 2012.

- Initial release.

W

wlc, [135](#)
wlc.config, [136](#)
wlc.main, [136](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component:slug)/, 108
ANY /, 86	GET /api/components/(string:project)/(string:component:slug)/, 110
/api	GET /api/components/(string:project)/(string:component:slug)/, 110
GET /api/, 89	GET /api/components/(string:project)/(string:component:slug)/, 109
/api/addons	GET /api/components/(string:project)/(string:component:slug)/, 108
GET /api/addons/, 124	GET /api/components/(string:project)/(string:component:slug)/, 112
GET /api/addons/(int:id)/, 124	GET /api/components/(string:project)/(string:component:slug)/, 111
PUT /api/addons/(int:id)/, 124	POST /api/components/(string:project)/(string:component:slug)/, 124
DELETE /api/addons/(int:id)/, 124	POST /api/components/(string:project)/(string:component:slug)/, 113
PATCH /api/addons/(int:id)/, 124	POST /api/components/(string:project)/(string:component:slug)/, 109
/api/changes	POST /api/components/(string:project)/(string:component:slug)/, 110
GET /api/changes/, 121	POST /api/components/(string:project)/(string:component:slug)/, 111
GET /api/changes/(int:id)/, 121	PUT /api/components/(string:project)/(string:component:slug)/, 107
/api/component-lists	DELETE /api/components/(string:project)/(string:component:slug)/, 108
GET /api/component-lists/, 125	DELETE /api/components/(string:project)/(string:component:slug)/, 113
GET /api/component-lists/(str:slug)/, 125	DELETE /api/components/(string:project)/(string:component:slug)/, 106
POST /api/component-lists/(str:slug)/components/, 125	/api/groups
PUT /api/component-lists/(str:slug)/, 125	GET /api/groups/, 92
DELETE /api/component-lists/(str:slug)/, 125	GET /api/groups/(int:id)/, 92
DELETE /api/component-lists/(str:slug)/components/(string:project)/, 126	POST /api/groups/, 92
PATCH /api/component-lists/(str:slug)/, 125	POST /api/groups/(int:id)/componentlists/, 95
/api/components	POST /api/groups/(int:id)/changes/, 94
GET /api/components/, 104	POST /api/groups/(int:id)/links/, 94
GET /api/components/(string:project)/(string:component:slug)/, 104	
GET /api/components/(string:project)/(string:component:slug)/changes/, 108	
GET /api/components/(string:project)/(string:component:slug)/links/, 112	

POST /api/groups/(int:id)/projects/, 94	/api/roles GET /api/roles/, 95
POST /api/groups/(int:id)/roles/, 94	GET /api/roles/(int:id)/, 95
PUT /api/groups/(int:id)/, 93	POST /api/roles/, 95
DELETE /api/groups/(int:id)/, 94	PUT /api/roles/(int:id)/, 96
DELETE /api/groups/(int:id)/componentlists/(int:component_list_id), 95	DELETE /api/roles/(int:id)/, 96
DELETE /api/groups/(int:id)/components/(int:component_id), 94	PATCH /api/roles/(int:id)/, 96
DELETE /api/groups/(int:id)/languages/(string:language_code), 95	/api/screenshots GET /api/screenshots/, 121
DELETE /api/groups/(int:id)/projects/(int:project_id), 94	GET /api/screenshots/(int:id)/, 121
PATCH /api/groups/(int:id)/, 93	GET /api/screenshots/(int:id)/file/, 122
	POST /api/screenshots/, 122
	POST /api/screenshots/(int:id)/file/, 122
	POST /api/screenshots/(int:id)/units/, 122
	PUT /api/screenshots/(int:id)/, 123
	DELETE /api/screenshots/(int:id)/, 123
	DELETE /api/screenshots/(int:id)/units/(int:unit_id), 122
	PATCH /api/screenshots/(int:id)/, 123
	/api/tasks GET /api/tasks/, 126
	GET /api/tasks/(str:uuid)/, 126
	/api/translations GET /api/translations/, 113
	GET /api/translations/(string:project)/(string:component), 113
	GET /api/translations/(string:project)/(string:component), 115
	GET /api/translations/(string:project)/(string:component), 117
	GET /api/translations/(string:project)/(string:component), 117
	GET /api/translations/(string:project)/(string:component), 118
	GET /api/translations/(string:project)/(string:component), 116
	POST /api/translations/(string:project)/(string:component), 116
	POST /api/translations/(string:project)/(string:component), 117
	POST /api/translations/(string:project)/(string:component), 118
	POST /api/translations/(string:project)/(string:component), 116
	DELETE /api/translations/(string:project)/(string:component), 115
	/api/units GET /api/units/, 119
	GET /api/units/(int:id)/, 119
	PUT /api/units/(int:id)/, 120
DELETE /api/projects/(string:project)/, 99	
PATCH /api/projects/(string:project)/, 99	

DELETE /api/units/(int:id)/, [120](#)
 PATCH /api/units/(int:id)/, [120](#)

/api/users

GET /api/users/, [89](#)
 GET /api/users/(str:username)/, [89](#)
 GET /api/users/(str:username)/notifications/,
[91](#)
 GET /api/users/(str:username)/notifications/(int:subscription_id)/,
[91](#)
 GET /api/users/(str:username)/statistics/,
[91](#)
 POST /api/users/, [89](#)
 POST /api/users/(str:username)/groups/,
[91](#)
 POST /api/users/(str:username)/notifications/,
[91](#)
 PUT /api/users/(str:username)/, [90](#)
 PUT /api/users/(str:username)/notifications/(int:subscription_id)/,
[91](#)
 DELETE /api/users/(str:username)/, [91](#)
 DELETE /api/users/(str:username)/notifications/(int:subscription_id)/,
[92](#)
 PATCH /api/users/(str:username)/, [90](#)
 PATCH /api/users/(str:username)/notifications/(int:subscription_id)/,
[92](#)

/exports

GET /exports/rss/, [130](#)
 GET /exports/rss/(string:project)/, [130](#)
 GET /exports/rss/(string:project)/(string:component)/,
[130](#)
 GET /exports/rss/(string:project)/(string:component)/(string:language)/,
[130](#)
 GET /exports/rss/language/(string:language)/,
[130](#)
 GET /exports/stats/(string:project)/(string:component)/,
[129](#)

/hooks

GET /hooks/update/(string:project)/,
[127](#)
 GET /hooks/update/(string:project)/(string:component)/,
[127](#)
 POST /hooks/azure/, [128](#)
 POST /hooks/bitbucket/, [127](#)
 POST /hooks/gitea/, [128](#)
 POST /hooks/gitee/, [128](#)
 POST /hooks/github/, [127](#)
 POST /hooks/gitlab/, [127](#)
 POST /hooks/pagure/, [128](#)

Símbolos

- .XML resource file
 - file format, [73](#)
- add
 - auto_translate command line option, [346](#)
- addon ADDON
 - install_addon command line option, [352](#)
- age HOURS
 - commit_pending command line option, [347](#)
- author USER@EXAMPLE.COM
 - add_suggestions command line option, [346](#)
- base-file-template TEMPLATE
 - import_project command line option, [350](#)
- check
 - importusers command line option, [352](#)
- config PATH
 - wlc command line option, [132](#)
- config-section SECTION
 - wlc command line option, [132](#)
- configuration CONFIG
 - install_addon command line option, [352](#)
- convert
 - wlc command line option, [133](#)
- email USER@EXAMPLE.COM
 - createadmin command line option, [348](#)
- file-format FORMAT
 - import_project command line option, [350](#)
- force
 - loadpo command line option, [353](#)
- force-commit
 - pushgit command line option, [354](#)
- format {csv,json,text,html}
 - wlc command line option, [132](#)
- ignore
 - import_json command line option, [349](#)
- inconsistent
 - auto_translate command line option, [346](#)
- input
 - wlc command line option, [133](#)
- key KEY
 - wlc command line option, [132](#)
- lang LANGUAGE
 - loadpo command line option, [353](#)
- language-code
 - list_translators command line option, [353](#)
- language-map LANGMAP
 - import_memory command line option, [350](#)
- language-regex REGEX
 - import_project command line option, [350](#)
- license NAME
 - import_project command line option, [351](#)
- license-url URL
 - import_project command line option, [351](#)
- main-component
 - import_project command line option, [350](#)
- main-component COMPONENT
 - import_json command line option, [349](#)
- mode MODE
 - auto_translate command line option, [346](#)
- mt MT
 - auto_translate command line option, [346](#)
- name
 - createadmin command line option, [348](#)
- name-template TEMPLATE
 - import_project command line option, [350](#)
- new-base-template TEMPLATE
 - import_project command line option, [350](#)
- no-password
 - createadmin command line option, [348](#)

--no-privs-update
 setupgroups command line option, 355
 --no-projects-update
 setupgroups command line option, 355
 --no-update
 setuplang command line option, 355
 --output
 wlc command line option, 133
 --overwrite
 auto_translate command line option, 346
 wlc command line option, 133
 --password PASSWORD
 createadmin command line option, 348
 --project PROJECT
 import_json command line option, 349
 --source PROJECT/COMPONENT
 auto_translate command line option, 346
 --threshold THRESHOLD
 auto_translate command line option, 346
 --update
 createadmin command line option, 348
 import_json command line option, 349
 install_addon command line option, 352
 --url URL
 wlc command line option, 132
 --user USERNAME
 auto_translate command line option, 346
 --username USERNAME
 createadmin command line option, 348
 --vcs NAME
 import_project command line option, 351

A

add_suggestions
 weblate admin command, 346
 add_suggestions command line option
 --author USER@EXAMPLE.COM, 346
 ADMINS
 setting, 181
 AKISMET_API_KEY
 setting, 300
 ALLOWED_HOSTS
 setting, 181
 Android
 file format, 68
 ANONYMOUS_USER_NAME
 setting, 301
 API, 86, 130, 135
 Apple strings
 file format, 69
 ARB
 file format, 72

AUDITLOG_EXPIRY
 setting, 301
 AUTH_LOCK_ATTEMPTS
 setting, 301
 AUTH_TOKEN_VALID
 setting, 302
 auto_translate
 weblate admin command, 346
 auto_translate command line option
 --add, 346
 --inconsistent, 346
 --mode MODE, 346
 --mt MT, 346
 --overwrite, 346
 --source PROJECT/COMPONENT, 346
 --threshold THRESHOLD, 346
 --user USERNAME, 346

AUTO_UPDATE
 setting, 301

AUTOFIX_LIST
 setting, 302

AVATAR_URL_PREFIX
 setting, 302

B

BACKGROUND_TASKS
 setting, 303

BASE_DIR
 setting, 303

BaseAddon (*classe em weblate.addons.base*), 390

BASIC_LANGUAGES
 setting, 303

bilingual
 translation, 60

C

can_install() (*método de classe weblate.addons.base.BaseAddon*), 390

CELERY_BACKUP_OPTIONS, 142, 156

CELERY_BEAT_OPTIONS, 142, 156

CELERY_MAIN_OPTIONS, 142, 156

CELERY_MEMORY_OPTIONS, 142, 156

CELERY_NOTIFY_OPTIONS, 142, 156

celery_queues
 weblate admin command, 347

CELERY_TRANSLATE_OPTIONS, 142, 156

changes
 wlc command line option, 133

CHECK_LIST
 setting, 304

checkgit
 weblate admin command, 347

cleanup
 wlc command line option, 133

cleanuptrans
 weblate admin command, 348

Comma separated values
 file format, 74

Command (*classe em wlc.main*), 137
COMMENT_CLEANUP_DAYS
 setting, 304
commit
 wlc command line option, 132
commit_pending
 weblate admin command, 347
commit_pending command line option
 --age HOURS, 347
COMMIT_PENDING_HOURS
 setting, 305
commitgit
 weblate admin command, 347
configure() (*método*
 blate.addons.base.BaseAddon), 390
CONTACT_FORM
 setting, 305
createadmin
 weblate admin command, 348
createadmin command line option
 --email USER@EXAMPLE.COM, 348
 --name, 348
 --no-password, 348
 --password PASSWORD, 348
 --update, 348
 --username USERNAME, 348
CSP_CONNECT_SRC
 setting, 304
CSP_FONT_SRC
 setting, 304
CSP_IMG_SRC
 setting, 304
CSP_SCRIPT_SRC
 setting, 304
CSP_STYLE_SRC
 setting, 304
CSV
 file format, 74

D

daily() (*método* *weblate.addons.base.BaseAddon*),
 390
DATA_DIR
 setting, 305
DATABASE_BACKUP
 setting, 306
DATABASES
 setting, 181
DEBUG
 setting, 182
DEFAULT_ACCESS_CONTROL
 setting, 306
DEFAULT_ADD_MESSAGE
 setting, 307
DEFAULT_ADDON_MESSAGE
 setting, 307
DEFAULT_ADDONS
 setting, 307

DEFAULT_AUTO_WATCH
 setting, 306
DEFAULT_COMMIT_MESSAGE
 setting, 307
DEFAULT_COMMITER_EMAIL
 setting, 307
DEFAULT_COMMITER_NAME
 setting, 307
DEFAULT_DELETE_MESSAGE
 setting, 307
DEFAULT_FROM_EMAIL
 setting, 182
DEFAULT_LANGUAGE
 setting, 308
DEFAULT_MERGE_MESSAGE
 setting, 307
DEFAULT_MERGE_STYLE
 setting, 308
DEFAULT_PAGE_LIMIT
 setting, 322
DEFAULT_PULL_MESSAGE
 setting, 308
DEFAULT_RESTRICTED_COMPONENT
 setting, 306
DEFAULT_SHARED_TM
 setting, 308
DEFAULT_TRANSLATION_PROPAGATION
 setting, 308
download
 wlc command line option, 133
DTD
 file format, 75
dump_memory
 weblate admin command, 348
dumpuserdata
 weblate admin command, 348

E

ENABLE_AVATARS
 setting, 309
ENABLE_HOOKS
 setting, 309
ENABLE_HTTPS
 setting, 309
ENABLE_SHARING
 setting, 309

F

file format
 .XML resource file, 73
 Android, 68
 Apple strings, 69
 ARB, 72
 Comma separated values, 74
 CSV, 74
 DTD, 75
 gettext, 62
 go-i18n, 72

- GWT properties, 66
- i18next, 71
- INI translations, 66, 67
- Java properties, 65
- Joomla translations, 67
- JSON, 70
- mil8n lang, 66
- PHP strings, 69
- PO, 62
- Qt, 68
- RC, 76
- RESX, 73
- Ruby YAML, 75
- Ruby YAML Ain't Markup Language, 75
- string resources, 68
- TS, 68
- XLIFF, 64
- XML, 76
- YAML, 74
- YAML Ain't Markup Language, 74

G

- `get()` (método *wlc.Weblate*), 136
- `get_add_form()` (método de classe *weblate.addons.base.BaseAddon*), 390
- `GET_HELP_URL`
 - setting, 309
- `get_settings_form()` (método *weblate.addons.base.BaseAddon*), 390
- `gettext`
 - file format, 62
- `GITHUB_CREDENTIALS`
 - setting, 310
- `GITHUB_TOKEN`
 - setting, 311
- `GITHUB_USERNAME`
 - setting, 310
- `GITLAB_CREDENTIALS`
 - setting, 309
- `GITLAB_TOKEN`
 - setting, 310
- `GITLAB_USERNAME`
 - setting, 310
- `go-i18n`
 - file format, 72
- `GOOGLE_ANALYTICS_ID`
 - setting, 311
- GWT properties
 - file format, 66

H

- `HIDE_REPO_CREDENTIALS`
 - setting, 311
- `HIDE_VERSION`
 - setting, 311

I

- i18next

- file format, 71
- `import_demo`
 - `weblate admin command`, 349
- `import_json`
 - `weblate admin command`, 349
- `import_json command line option`
 - `--ignore`, 349
 - `--main-component COMPONENT`, 349
 - `--project PROJECT`, 349
 - `--update`, 349
- `import_memory`
 - `weblate admin command`, 350
- `import_memory command line option`
 - `--language-map LANGMAP`, 350
- `import_project`
 - `weblate admin command`, 350
- `import_project command line option`
 - `--base-file-template TEMPLATE`, 350
 - `--file-format FORMAT`, 350
 - `--language-regex REGEX`, 350
 - `--license NAME`, 351
 - `--license-url URL`, 351
 - `--main-component`, 350
 - `--name-template TEMPLATE`, 350
 - `--new-base-template TEMPLATE`, 350
 - `--vcs NAME`, 351
- `importuserdata`
 - `weblate admin command`, 352
- `importusers`
 - `weblate admin command`, 352
- `importusers command line option`
 - `--check`, 352
- INI translations
 - file format, 66, 67
- `install_addon`
 - `weblate admin command`, 352
- `install_addon command line option`
 - `--addon ADDON`, 352
 - `--configuration CONFIG`, 352
 - `--update`, 352
- `IP_BEHIND_REVERSE_PROXY`
 - setting, 311
- `IP_PROXY_HEADER`
 - setting, 312
- `IP_PROXY_OFFSET`
 - setting, 312
- iPad
 - translation, 69
- iPhone
 - translation, 69

J

- Java properties
 - file format, 65
- Joomla translations
 - file format, 67
- JSON
 - file format, 70

L

`LEGAL_URL`
 setting, 312

`LICENSE_EXTRA`
 setting, 313

`LICENSE_FILTER`
 setting, 313

`LICENSE_REQUIRED`
 setting, 313

`LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH`
 setting, 314

`list_languages`
 weblate admin command, 353

`list_translators`
 weblate admin command, 353

`list_translators` command line option
 --language-code, 353

`list_versions`
 weblate admin command, 353

`list-components`
 wlc command line option, 132

`list-languages`
 wlc command line option, 132

`list-projects`
 wlc command line option, 132

`list-translations`
 wlc command line option, 132

`load()` (*método* `wlc.config.WeblateConfig`), 136

`loadpo`
 weblate admin command, 353

`loadpo` command line option
 --force, 353
 --lang LANGUAGE, 353

`LOCALIZE_CDN_PATH`
 setting, 314

`LOCALIZE_CDN_URL`
 setting, 314

`lock`
 wlc command line option, 133

`lock_translation`
 weblate admin command, 354

`lock-status`
 wlc command line option, 133

`LOGIN_REQUIRED_URLS`
 setting, 314

`LOGIN_REQUIRED_URLS_EXCEPTIONS`
 setting, 314

`ls`
 wlc command line option, 132

M

`MACHINE_TRANSLATION_SERVICES`
 setting, 315

`main()` (*no módulo* `wlc.main`), 136

`MATOMO_SITE_ID`
 setting, 315

`MATOMO_URL`
 setting, 315

`mi18n lang`
 file format, 66

`monolingual`
 translation, 60

`move_language`
 weblate admin command, 354

`MT_APERTIUM_APY`
 setting, 316

`MT_AWS_ACCESS_KEY_ID`
 setting, 316

`MT_AWS_REGION`
 setting, 316

`MT_AWS_SECRET_ACCESS_KEY`
 setting, 316

`MT_BAIDU_ID`
 setting, 316

`MT_BAIDU_SECRET`
 setting, 317

`MT_DEEPL_API_URL`
 setting, 317

`MT_DEEPL_KEY`
 setting, 317

`MT_GOOGLE_CREDENTIALS`
 setting, 318

`MT_GOOGLE_KEY`
 setting, 318

`MT_GOOGLE_LOCATION`
 setting, 318

`MT_GOOGLE_PROJECT`
 setting, 318

`MT_LIBRETRANSLATE_API_URL`
 setting, 317

`MT_LIBRETRANSLATE_KEY`
 setting, 318

`MT_MICROSOFT_BASE_URL`
 setting, 318

`MT_MICROSOFT_COGNITIVE_KEY`
 setting, 319

`MT_MICROSOFT_ENDPOINT_URL`
 setting, 319

`MT_MICROSOFT_REGION`
 setting, 319

`MT_MODERNMT_KEY`
 setting, 319

`MT_MODERNMT_URL`
 setting, 319

`MT_MYMEMORY_EMAIL`
 setting, 319

`MT_MYMEMORY_KEY`
 setting, 319

`MT_MYMEMORY_USER`
 setting, 320

`MT_NETEASE_KEY`
 setting, 320

`MT_NETEASE_SECRET`
 setting, 320

`MT_SAP_BASE_URL`
 setting, 321

MT_SAP_PASSWORD
 setting, 321
 MT_SAP_SANDBOX_APIKEY
 setting, 321
 MT_SAP_USE_MT
 setting, 321
 MT_SAP_USERNAME
 setting, 321
 MT_SERVICES
 setting, 315
 MT_TMSERVER
 setting, 320
 MT_YANDEX_KEY
 setting, 320
 MT_YOUDAO_ID
 setting, 320
 MT_YOUDAO_SECRET
 setting, 320
 módulo
 wlc, 135
 wlc.config, 136
 wlc.main, 136

N

NEARBY_MESSAGES
 setting, 321

P

PAGURE_CREDENTIALS
 setting, 322
 PAGURE_TOKEN
 setting, 322
 PAGURE_USERNAME
 setting, 322
 PHP strings
 file format, 69
 PIWIK_SITE_ID
 setting, 315
 PIWIK_URL
 setting, 315
 PO
 file format, 62
 post() (método wlc.Weblate), 136
 post_add() (método
 blate.addons.base.BaseAddon), 390
 post_commit() (método
 blate.addons.base.BaseAddon), 390
 post_push() (método
 blate.addons.base.BaseAddon), 390
 post_update() (método
 blate.addons.base.BaseAddon), 390
 pre_commit() (método
 blate.addons.base.BaseAddon), 390
 pre_push() (método
 blate.addons.base.BaseAddon), 390
 pre_update() (método
 blate.addons.base.BaseAddon), 390
 pull

 wlc command line option, 132
 push
 wlc command line option, 133
 pushgit
 weblate admin command, 354
 pushgit command line option
 --force-commit, 354
 Python, 135

Q

Qt
 file format, 68

R

RATELIMIT_ATTEMPTS
 setting, 322
 RATELIMIT_LOCKOUT
 setting, 323
 RATELIMIT_WINDOW
 setting, 323
 RC
 file format, 76
 register_command() (no módulo wlc.main), 137
 REGISTRATION_ALLOW_BACKENDS
 setting, 323
 REGISTRATION_CAPTCHA
 setting, 323
 REGISTRATION_EMAIL_MATCH
 setting, 324
 REGISTRATION_OPEN
 setting, 324
 repo
 wlc command line option, 133
 REPOSITORY_ALERT_THRESHOLD
 setting, 324
 REQUIRE_LOGIN
 setting, 324
 reset
 wlc command line option, 133
 REST, 86
 RESX
 file format, 73
 RFC
 RFC 5646, 59
 we- Ruby YAML
 file format, 75
 we- Ruby YAML Ain't Markup Language
 file format, 75

S

we- save_state() (método
 blate.addons.base.BaseAddon), 390
 SECRET_KEY
 setting, 182
 SENTRY_DSN
 setting, 325
 we- SERVER_EMAIL
 setting, 182

SESSION_COOKIE_AGE_AUTHENTICATED
 setting, 325
 SESSION_ENGINE
 setting, 181
 setting
 ADMINS, 181
 AKISMET_API_KEY, 300
 ALLOWED_HOSTS, 181
 ANONYMOUS_USER_NAME, 301
 AUDITLOG_EXPIRY, 301
 AUTH_LOCK_ATTEMPTS, 301
 AUTH_TOKEN_VALID, 302
 AUTO_UPDATE, 301
 AUTOFIX_LIST, 302
 AVATAR_URL_PREFIX, 302
 BACKGROUND_TASKS, 303
 BASE_DIR, 303
 BASIC_LANGUAGES, 303
 CHECK_LIST, 304
 COMMENT_CLEANUP_DAYS, 304
 COMMIT_PENDING_HOURS, 305
 CONTACT_FORM, 305
 CSP_CONNECT_SRC, 304
 CSP_FONT_SRC, 304
 CSP_IMG_SRC, 304
 CSP_SCRIPT_SRC, 304
 CSP_STYLE_SRC, 304
 DATA_DIR, 305
 DATABASE_BACKUP, 306
 DATABASES, 181
 DEBUG, 182
 DEFAULT_ACCESS_CONTROL, 306
 DEFAULT_ADD_MESSAGE, 307
 DEFAULT_ADDON_MESSAGE, 307
 DEFAULT_ADDONS, 307
 DEFAULT_AUTO_WATCH, 306
 DEFAULT_COMMIT_MESSAGE, 307
 DEFAULT_COMMITER_EMAIL, 307
 DEFAULT_COMMITER_NAME, 307
 DEFAULT_DELETE_MESSAGE, 307
 DEFAULT_FROM_EMAIL, 182
 DEFAULT_LANGUAGE, 308
 DEFAULT_MERGE_MESSAGE, 307
 DEFAULT_MERGE_STYLE, 308
 DEFAULT_PAGE_LIMIT, 322
 DEFAULT_PULL_MESSAGE, 308
 DEFAULT_RESTRICTED_COMPONENT, 306
 DEFAULT_SHARED_TM, 308
 DEFAULT_TRANSLATION_PROPAGATION,
 308
 ENABLE_AVATARS, 309
 ENABLE_HOOKS, 309
 ENABLE_HTTPS, 309
 ENABLE_SHARING, 309
 GET_HELP_URL, 309
 GITHUB_CREDENTIALS, 310
 GITHUB_TOKEN, 311
 GITHUB_USERNAME, 310
 GITLAB_CREDENTIALS, 309
 GITLAB_TOKEN, 310
 GITLAB_USERNAME, 310
 GOOGLE_ANALYTICS_ID, 311
 HIDE_REPO_CREDENTIALS, 311
 HIDE_VERSION, 311
 IP_BEHIND_REVERSE_PROXY, 311
 IP_PROXY_HEADER, 312
 IP_PROXY_OFFSET, 312
 LEGAL_URL, 312
 LICENSE_EXTRA, 313
 LICENSE_FILTER, 313
 LICENSE_REQUIRED, 313
 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH,
 314
 LOCALIZE_CDN_PATH, 314
 LOCALIZE_CDN_URL, 314
 LOGIN_REQUIRED_URLS, 314
 LOGIN_REQUIRED_URLS_EXCEPTIONS, 314
 MACHINE_TRANSLATION_SERVICES, 315
 MATOMO_SITE_ID, 315
 MATOMO_URL, 315
 MT_APERTIUM_API, 316
 MT_AWS_ACCESS_KEY_ID, 316
 MT_AWS_REGION, 316
 MT_AWS_SECRET_ACCESS_KEY, 316
 MT_BAIDU_ID, 316
 MT_BAIDU_SECRET, 317
 MT_DEEPL_API_URL, 317
 MT_DEEPL_KEY, 317
 MT_GOOGLE_CREDENTIALS, 318
 MT_GOOGLE_KEY, 318
 MT_GOOGLE_LOCATION, 318
 MT_GOOGLE_PROJECT, 318
 MT_LIBRETRANSLATE_API_URL, 317
 MT_LIBRETRANSLATE_KEY, 318
 MT_MICROSOFT_BASE_URL, 318
 MT_MICROSOFT_COGNITIVE_KEY, 319
 MT_MICROSOFT_ENDPOINT_URL, 319
 MT_MICROSOFT_REGION, 319
 MT_MODERNMT_KEY, 319
 MT_MODERNMT_URL, 319
 MT_MYMEMORY_EMAIL, 319
 MT_MYMEMORY_KEY, 319
 MT_MYMEMORY_USER, 320
 MT_NETEASE_KEY, 320
 MT_NETEASE_SECRET, 320
 MT_SAP_BASE_URL, 321
 MT_SAP_PASSWORD, 321
 MT_SAP_SANDBOX_APIKEY, 321
 MT_SAP_USE_MT, 321
 MT_SAP_USERNAME, 321
 MT_SERVICES, 315
 MT_TMSEVER, 320
 MT_YANDEX_KEY, 320
 MT_YOUDAO_ID, 320
 MT_YOUDAO_SECRET, 320
 NEARBY_MESSAGES, 321

- PAGURE_CREDENTIALS, 322
- PAGURE_TOKEN, 322
- PAGURE_USERNAME, 322
- PIWIK_SITE_ID, 315
- PIWIK_URL, 315
- RATELIMIT_ATTEMPTS, 322
- RATELIMIT_LOCKOUT, 323
- RATELIMIT_WINDOW, 323
- REGISTRATION_ALLOW_BACKENDS, 323
- REGISTRATION_CAPTCHA, 323
- REGISTRATION_EMAIL_MATCH, 324
- REGISTRATION_OPEN, 324
- REPOSITORY_ALERT_THRESHOLD, 324
- REQUIRE_LOGIN, 324
- SECRET_KEY, 182
- SENTRY_DSN, 325
- SERVER_EMAIL, 182
- SESSION_COOKIE_AGE_AUTHENTICATED, 325
- SESSION_ENGINE, 181
- SIMPLIFY_LANGUAGES, 325
- SINGLE_PROJECT, 326
- SITE_DOMAIN, 325
- SITE_TITLE, 326
- SPECIAL_CHARS, 326
- STATUS_URL, 326
- SUGGESTION_CLEANUP_DAYS, 326
- UPDATE_LANGUAGES, 326
- URL_PREFIX, 327
- VCS_BACKENDS, 327
- VCS_CLONE_DEPTH, 327
- WEBLATE_ADDONS, 328
- WEBLATE_EXPORTERS, 329
- WEBLATE_FORMATS, 329
- WEBLATE_GPG_IDENTITY, 329
- WEBSITE_REQUIRED, 329
- setupgroups
 - weblate admin command, 355
- setupgroups command line option
 - no-privs-update, 355
 - no-projects-update, 355
- setuplang
 - weblate admin command, 355
- setuplang command line option
 - no-update, 355
- show
 - wlc command line option, 132
- SIMPLIFY_LANGUAGES
 - setting, 325
- SINGLE_PROJECT
 - setting, 326
- SITE_DOMAIN
 - setting, 325
- SITE_TITLE
 - setting, 326
- SPECIAL_CHARS
 - setting, 326
- statistics
 - wlc command line option, 133
- STATUS_URL
 - setting, 326
- stay_on_create
 - (atributo *blate.addons.base.BaseAddon*), 390
- store_post_load()
 - (método *blate.addons.base.BaseAddon*), 391
- string resources
 - file format, 68
- SUGGESTION_CLEANUP_DAYS
 - setting, 326
- T
- translation
 - bilingual, 60
 - iPad, 69
 - iPhone, 69
 - monolingual, 60
- TS
 - file format, 68
- U
- unit_pre_create()
 - (método *blate.addons.base.BaseAddon*), 391
- unlock
 - wlc command line option, 133
- unlock_translation
 - weblate admin command, 355
- UPDATE_LANGUAGES
 - setting, 326
- updatechecks
 - weblate admin command, 355
- updategit
 - weblate admin command, 356
- upload
 - wlc command line option, 133
- URL_PREFIX
 - setting, 327
- UWSGI_WORKERS, 142, 156
- V
- variável de ambiente
 - CELERY_BACKUP_OPTIONS, 142, 156
 - CELERY_BEAT_OPTIONS, 142, 156
 - CELERY_MAIN_OPTIONS, 142, 156
 - CELERY_MEMORY_OPTIONS, 142, 156
 - CELERY_NOTIFY_OPTIONS, 142, 156
 - CELERY_TRANSLATE_OPTIONS, 142, 156
 - POSTGRES_ALTER_ROLE, 153
 - POSTGRES_DATABASE, 152
 - POSTGRES_HOST, 152
 - POSTGRES_PASSWORD, 152
 - POSTGRES_PASSWORD_FILE, 152
 - POSTGRES_PORT, 152
 - POSTGRES_SSL_MODE, 153
 - POSTGRES_USER, 152
 - REDIS_DB, 153
 - REDIS_HOST, 153

- REDIS_PASSWORD, 153
- REDIS_PORT, 153
- REDIS_TLS, 153
- REDIS_VERIFY_SSL, 153
- ROLLBAR_ENVIRONMENT, 155
- ROLLBAR_KEY, 155
- SENTRY_DSN, 155
- SENTRY_ENVIRONMENT, 155
- SOCIAL_AUTH_SLACK_SECRET, 152
- UWSGI_WORKERS, 142, 156, 157
- WEBLATE_ADD_ADDONS, 156
- WEBLATE_ADD_APPS, 156
- WEBLATE_ADD_AUTOFIX, 156
- WEBLATE_ADD_CHECK, 156
- WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 146
- WEBLATE_ADMIN_EMAIL, 142--144, 148
- WEBLATE_ADMIN_NAME, 142--144
- WEBLATE_ADMIN_PASSWORD, 139, 142--144
- WEBLATE_ADMIN_PASSWORD_FILE, 144
- WEBLATE_AKISMET_API_KEY, 147, 363
- WEBLATE_ALLOWED_HOSTS, 144, 181, 186, 326
- WEBLATE_AUTH_LDAP_BIND_DN, 149
- WEBLATE_AUTH_LDAP_BIND_PASSWORD, 149
- WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERENCE, 149
- WEBLATE_AUTH_LDAP_SERVER_URI, 149
- WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 149
- WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 149
- WEBLATE_AUTH_LDAP_USER_SEARCH, 149
- WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 149
- WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 149
- WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELETE, 149
- WEBLATE_BASIC_LANGUAGES, 147
- WEBLATE_CONTACT_FORM, 144
- WEBLATE_CSP_CONNECT_SRC, 147
- WEBLATE_CSP_FONT_SRC, 147
- WEBLATE_CSP_IMG_SRC, 147
- WEBLATE_CSP_SCRIPT_SRC, 147
- WEBLATE_CSP_STYLE_SRC, 147
- WEBLATE_DATABASE_BACKUP, 153
- WEBLATE_DEBUG, 143
- WEBLATE_DEFAULT_ACCESS_CONTROL, 146
- WEBLATE_DEFAULT_AUTO_WATCH, 147
- WEBLATE_DEFAULT_COMMITER_EMAIL, 147
- WEBLATE_DEFAULT_COMMITER_NAME, 147
- WEBLATE_DEFAULT_FROM_EMAIL, 144
- WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 146
- WEBLATE_DEFAULT_SHARED_TM, 147
- WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 146
- WEBLATE_EMAIL_BACKEND, 154
- WEBLATE_EMAIL_HOST, 154
- WEBLATE_EMAIL_HOST_PASSWORD, 154
- WEBLATE_EMAIL_HOST_PASSWORD_FILE, 154
- WEBLATE_EMAIL_HOST_USER, 154
- WEBLATE_EMAIL_PORT, 154
- WEBLATE_EMAIL_USE_SSL, 154
- WEBLATE_EMAIL_USE_TLS, 154
- WEBLATE_ENABLE_AVATARS, 148
- WEBLATE_ENABLE_HTTPS, 144, 213
- WEBLATE_GET_HELP_URL, 155
- WEBLATE_GITHUB_TOKEN, 146
- WEBLATE_GITHUB_USERNAME, 146
- WEBLATE_GITLAB_TOKEN, 146
- WEBLATE_GITLAB_USERNAME, 146
- WEBLATE_GOOGLE_ANALYTICS_ID, 146
- WEBLATE_GPG_IDENTITY, 147
- WEBLATE_HIDE_VERSION, 147
- WEBLATE_IP_PROXY_HEADER, 145
- WEBLATE_LEGAL_URL, 155
- WEBLATE_LICENSE_FILTER, 147
- WEBLATE_LICENSE_REQUIRED, 147
- WEBLATE_LOCALIZE_CDN_PATH, 155
- WEBLATE_LOCALIZE_CDN_URL, 155
- WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 145
- WEBLATE_LOGLEVEL, 143
- WEBLATE_MT_APERTIUM_API_KEY, 148
- WEBLATE_MT_AWS_ACCESS_KEY_ID, 148
- WEBLATE_MT_AWS_REGION, 148
- WEBLATE_MT_AWS_SECRET_ACCESS_KEY, 148
- WEBLATE_MT_DEEPL_API_URL, 148
- WEBLATE_MT_DEEPL_KEY, 148
- WEBLATE_MT_GLOSBE_ENABLED, 149
- WEBLATE_MT_GOOGLE_KEY, 148
- WEBLATE_MT_LIBRETRANSLATE_API_URL, 148
- WEBLATE_MT_LIBRETRANSLATE_KEY, 148
- WEBLATE_MT_MICROSOFT_BASE_URL, 148
- WEBLATE_MT_MICROSOFT_COGNITIVE_KEY, 148
- WEBLATE_MT_MICROSOFT_ENDPOINT_URL, 148
- WEBLATE_MT_MICROSOFT_REGION, 148
- WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED, 149
- WEBLATE_MT_MODERNMT_KEY, 148
- WEBLATE_MT_MYMONEY_ENABLED, 148
- WEBLATE_MT_SAP_BASE_URL, 149
- WEBLATE_MT_SAP_PASSWORD, 149
- WEBLATE_MT_SAP_SANDBOX_APIKEY, 149
- WEBLATE_MT_SAP_USE_MT, 149
- WEBLATE_MT_SAP_USERNAME, 149
- WEBLATE_NO_EMAIL_AUTH, 152

- WEBLATE_PAGURE_TOKEN, [146](#)
- WEBLATE_PAGURE_USERNAME, [146](#)
- WEBLATE_RATELIMIT_ATTEMPTS, [147](#), [365](#)
- WEBLATE_RATELIMIT_LOCKOUT, [147](#)
- WEBLATE_RATELIMIT_WINDOW, [147](#)
- WEBLATE_REGISTRATION_ALLOW_BACKENDS, [144](#)
- WEBLATE_REGISTRATION_OPEN, [144](#)
- WEBLATE_REMOVE_ADDONS, [156](#)
- WEBLATE_REMOVE_APPS, [156](#)
- WEBLATE_REMOVE_AUTOFIX, [156](#)
- WEBLATE_REMOVE_CHECK, [156](#)
- WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTED, [146](#)
- WEBLATE_REQUIRE_LOGIN, [145](#), [325](#)
- WEBLATE_SAML_IDP_ENTITY_ID, [152](#)
- WEBLATE_SAML_IDP_URL, [152](#)
- WEBLATE_SAML_IDP_X509CERT, [152](#)
- WEBLATE_SECURE_PROXY_SSL_HEADER, [145](#)
- WEBLATE_SERVER_EMAIL, [144](#)
- WEBLATE_SERVICE, [142](#), [157](#)
- WEBLATE_SILENCED_SYSTEM_CHECKS, [147](#), [210](#)
- WEBLATE_SIMPLIFY_LANGUAGES, [146](#)
- WEBLATE_SITE_DOMAIN, [143](#), [183](#), [202](#), [326](#)
- WEBLATE_SITE_TITLE, [143](#)
- WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, [151](#)
- WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, [151](#)
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, [151](#)
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, [151](#)
- WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, [151](#)
- WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, [150](#)
- WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, [150](#)
- WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, [151](#)
- WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, [151](#)
- WEBLATE_SOCIAL_AUTH_FEDORA, [152](#)
- WEBLATE_SOCIAL_AUTH_GITHUB_KEY, [150](#)
- WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, [150](#)
- WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, [151](#)
- WEBLATE_SOCIAL_AUTH_GITLAB_KEY, [151](#)
- WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, [151](#)
- WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, [151](#)
- WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, [151](#)
- WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, [151](#)
- WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, [151](#)
- WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, [151](#)
- WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, [151](#)
- WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, [151](#)
- WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, [151](#)
- WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, [151](#)
- WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, [151](#)
- WEBLATE_SOCIAL_AUTH_OPENSUSE, [152](#)
- WEBLATE_SOCIAL_AUTH_SLACK_KEY, [152](#)
- WEBLATE_SOCIAL_AUTH_UBUNTU, [152](#)
- WEBLATE_STATUS_URL, [155](#)
- WEBLATE_TIME_ZONE, [144](#)
- WEBLATE_URL_PREFIX, [147](#)
- WEBLATE_WEBSITE_REQUIRED, [147](#)
- WEBLATE_WORKERS, [142](#), [156](#), [157](#)
- WL_BRANCH, [297](#)
- WL_COMPONENT_NAME, [298](#)
- WL_COMPONENT_SLUG, [298](#)
- WL_COMPONENT_URL, [298](#)
- WL_ENGAGE_URL, [298](#)
- WL_FILE_FORMAT, [298](#)
- WL_FILEMASK, [297](#)
- WL_LANGUAGE, [298](#)
- WL_NEW_BASE, [297](#)
- WL_PATCH, [297](#)
- WL_PREVIOUS_HEAD, [298](#)
- WL_PRODUCT_NAME, [298](#)
- WL_PROJECT_SLUG, [298](#)
- WL_REPO, [297](#)
- WL_TEMPLATE, [297](#)
- WL_VCS, [297](#)
- VCS_BACKENDS
 - setting, [327](#)
- VCS_CLONE_DEPTH
 - setting, [327](#)
- version
 - wlc command line option, [132](#)

W

- Weblate (*classe em wlc*), [136](#)
- weblate admin command
 - add_suggestions, [346](#)
 - auto_translate, [346](#)
 - celery_queues, [347](#)
 - checkgit, [347](#)
 - cleanup_trans, [348](#)
 - commit_pending, [347](#)
 - commitgit, [347](#)
 - createadmin, [348](#)

- dump_memory, 348
 - dumpuserdata, 348
 - import_demo, 349
 - import_json, 349
 - import_memory, 350
 - import_project, 350
 - importuserdata, 352
 - importusers, 352
 - install_addon, 352
 - list_languages, 353
 - list_translators, 353
 - list_versions, 353
 - loadpo, 353
 - lock_translation, 354
 - move_language, 354
 - pushgit, 354
 - setupgroups, 355
 - setuplang, 355
 - unlock_translation, 355
 - updatechecks, 355
 - updategit, 356
 - WEBLATE_ADDONS
 - setting, 328
 - WEBLATE_ADMIN_EMAIL, 142--144, 148
 - WEBLATE_ADMIN_NAME, 142--144
 - WEBLATE_ADMIN_PASSWORD, 139, 142--144
 - WEBLATE_ADMIN_PASSWORD_FILE, 144
 - WEBLATE_AKISMET_API_KEY, 363
 - WEBLATE_ALLOWED_HOSTS, 181, 186, 326
 - WEBLATE_EMAIL_HOST_PASSWORD, 154
 - WEBLATE_EMAIL_PORT, 154
 - WEBLATE_EMAIL_USE_SSL, 154
 - WEBLATE_EMAIL_USE_TLS, 154
 - WEBLATE_ENABLE_HTTPS, 213
 - WEBLATE_EXPORTERS
 - setting, 329
 - WEBLATE_FORMATS
 - setting, 329
 - WEBLATE_GPG_IDENTITY
 - setting, 329
 - WEBLATE_LOCALIZE_CDN_PATH, 155
 - WEBLATE_RATELIMIT_ATTEMPTS, 365
 - WEBLATE_REQUIRE_LOGIN, 325
 - WEBLATE_SECURE_PROXY_SSL_HEADER, 145
 - WEBLATE_SERVICE, 142
 - WEBLATE_SILENCED_SYSTEM_CHECKS, 210
 - WEBLATE_SITE_DOMAIN, 183, 202, 326
 - WEBLATE_WORKERS, 142, 156, 157
 - WeblateConfig (*classe em wlc.config*), 136
 - WeblateException, 135
 - WEBSITE_REQUIRED
 - setting, 329
 - wlc, 130
 - módulo, 135
 - wlc command line option
 - config PATH, 132
 - config-section SECTION, 132
 - convert, 133
 - format {csv,json,text,html}, 132
 - input, 133
 - key KEY, 132
 - output, 133
 - overwrite, 133
 - url URL, 132
 - changes, 133
 - cleanup, 133
 - commit, 132
 - download, 133
 - list-components, 132
 - list-languages, 132
 - list-projects, 132
 - list-translations, 132
 - lock, 133
 - lock-status, 133
 - ls, 132
 - pull, 132
 - push, 133
 - repo, 133
 - reset, 133
 - show, 132
 - statistics, 133
 - unlock, 133
 - upload, 133
 - version, 132
 - wlc.config
 - módulo, 136
 - wlc.main
 - módulo, 136
- ## X
- XLIFF
 - file format, 64
 - XML
 - file format, 76
- ## Y
- YAML
 - file format, 74
 - YAML Ain't Markup Language
 - file format, 74