



The Weblate Manual

Release 4.6.1

Michal Čihař

02 mai. 2021

1	Documentação de usuário	1
1.1	Básico do Weblate	1
1.2	Registro e perfil de usuário	1
1.3	Traduzindo usando o Weblate	10
1.4	Baixando e enviando traduções	20
1.5	Glossário	24
1.6	Verificações e correções	26
1.7	Pesquisando	43
1.8	Fluxos de trabalho de tradução	48
1.9	Perguntas frequentes	51
1.10	Formatos de arquivos suportados	59
1.11	Integração com controle de versão	79
1.12	API REST do Weblate	86
1.13	Weblate Client	130
1.14	API Python do Weblate	135
2	Documentação de administrador	137
2.1	Instruções de configuração	137
2.2	Implantações de Weblate	198
2.3	Atualizando o Weblate	199
2.4	Fazendo backup e movendo o Weblate	205
2.5	Autenticação	211
2.6	Controle de acesso	220
2.7	Projetos de tradução	230
2.8	Definições de idioma	247
2.9	Localização contínua	250
2.10	Licenciando traduções	259
2.11	Processo de tradução	260
2.12	Verificações e correções	266
2.13	Tradução de máquina	274
2.14	Extensões	280
2.15	Memória de tradução	291
2.16	Configuração	293
2.17	Amostra de configuração	321
2.18	Comandos de gerência	337
2.19	Anúncios	347
2.20	Lista de componentes	350
2.21	Módulos opcionais do Weblate	351
2.22	Personalizando o Weblate	356
2.23	Interface de gerenciamento	358
2.24	Obtendo suporte para o Weblate	366

2.25	Documentos legais	369
3	Documentação de colaborador	371
3.1	Contribuindo para o Weblate	371
3.2	Começando a contribuir código para o Weblate	373
3.3	Código-fonte do Weblate	377
3.4	Depuração do Weblate	378
3.5	Componentes internos do Weblate	379
3.6	Desenvolvendo extensões	381
3.7	Frontend do Weblate	383
3.8	Relatando problemas no Weblate	384
3.9	Conjunto de testes e integração contínua do Weblate	384
3.10	Esquemas de dados	386
3.11	Criando lançamento do Weblate	390
3.12	Segurança e privacidade	391
3.13	Sobre o Weblate	391
3.14	Licença	392
4	Histórico de alterações	393
4.1	Weblate 4.6.1	393
4.2	Weblate 4.6	393
4.3	Weblate 4.5.3	394
4.4	Weblate 4.5.2	394
4.5	Weblate 4.5.1	395
4.6	Weblate 4.5	395
4.7	Weblate 4.4.2	396
4.8	Weblate 4.4.1	396
4.9	Weblate 4.4	396
4.10	Weblate 4.3.2	397
4.11	Weblate 4.3.1	398
4.12	Weblate 4.3	398
4.13	Weblate 4.2.2	399
4.14	Weblate 4.2.1	399
4.15	Weblate 4.2	399
4.16	Weblate 4.1.1	400
4.17	Weblate 4.1	400
4.18	Weblate 4.0.4	402
4.19	Weblate 4.0.3	402
4.20	Weblate 4.0.2	402
4.21	Weblate 4.0.1	403
4.22	Weblate 4.0	403
4.23	Weblate 3.x series	404
4.24	Weblate 2.x series	415
4.25	Weblate 1.x series	426
4.26	Weblate 0.x series	430
	Índice de Módulos Python	434
	HTTP Routing Table	435
	Índice	438

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 arquivo traduzível (por exemplo, *GNU gettext* ou *Recurso de textos de Android*). Os projetos estão lá para ajudá-lo a organizar componentes em conjuntos lógicos (por exemplo, para agrupar todas as traduções usadas dentro de um aplicativo).

Internamente, cada projeto tem traduções para textos comuns propagados em outros componentes dentro dele por padrão. Isso alivia o fardo da tradução repetitiva e de várias versões. A propagação da tradução pode ser desabilitada por *Configuração de componente* usando *Permitir propagação de tradução* caso as traduções devam divergir.

Ver também:

`../devel/integration`

1.2 Registro e perfil de usuário

1.2.1 Registro

Todos podem procurar projetos, visualizar traduções ou sugerir traduções por padrão. Somente usuários registrados têm permissão para realmente salvar as alterações e são creditados para cada tradução feita.

Você pode se registrar seguindo alguns passos simples:

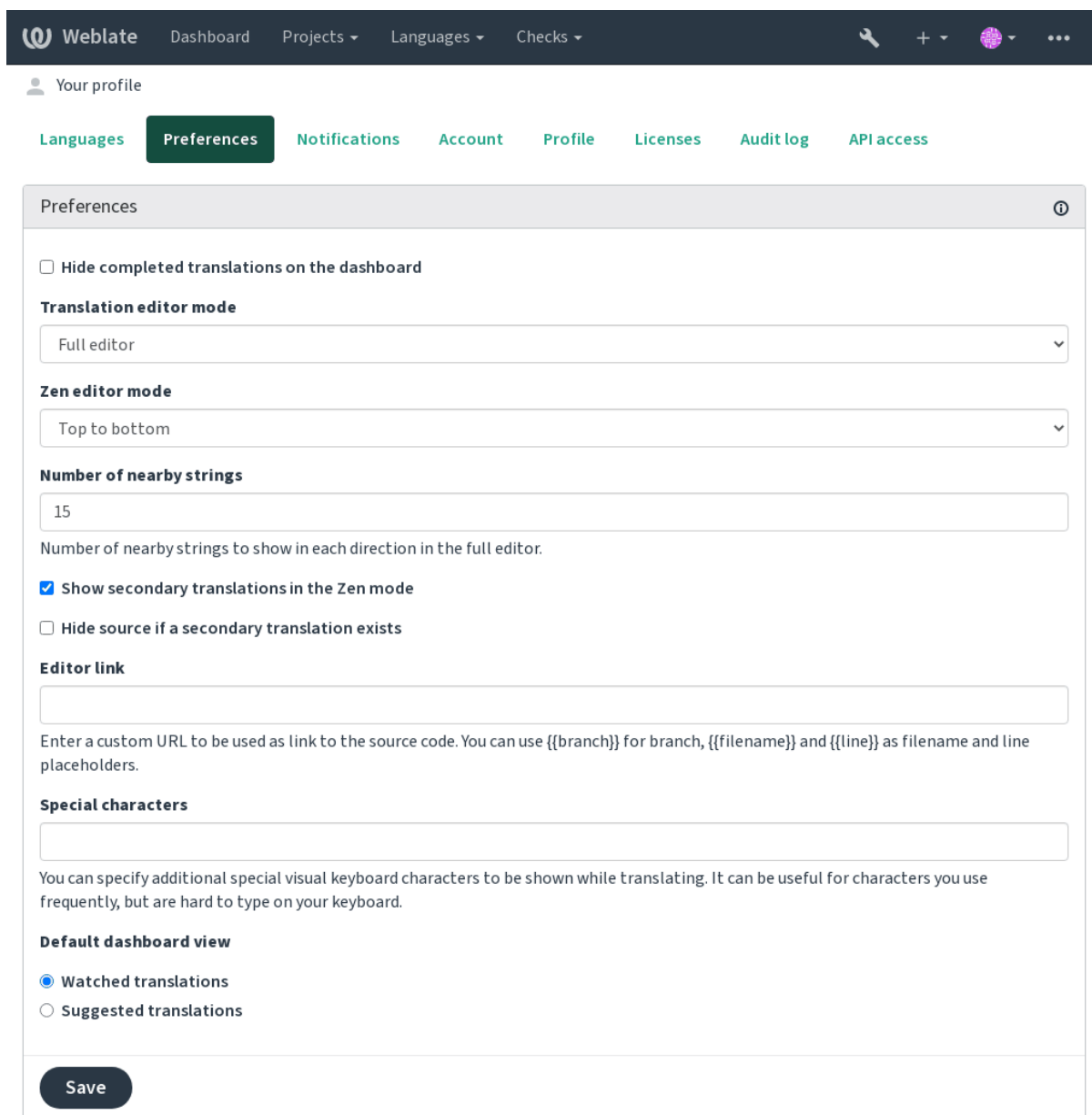
1. Preencha o formulário de registro com suas credenciais.
2. Ative o registro seguindo o link no e-mail que você receber.
3. Ajuste opcionalmente seu perfil para escolher quais idiomas você conhece.

1.2.2 Painei

Ao fazer login, você verá uma visão geral de projetos e componentes, bem como sua respectiva progressão de tradução. Novo na versão 2.5.

Os componentes dos projetos que você está observando são mostrados por padrão, e cruzados com os idiomas de sua preferência.

Dica: Você pode mudar para visualizações diferentes usando as abas de navegação.



The screenshot shows the Weblate user interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks' menus. On the right are icons for settings, a plus sign, a user profile, and a menu. Below this 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: a checkbox for 'Hide completed translations on the dashboard'; a 'Translation editor mode' dropdown set to 'Full editor'; a 'Zen editor mode' dropdown set to 'Top to bottom'; a text input for 'Number of nearby strings' set to '15' with a description; a checked checkbox for 'Show secondary translations in the Zen mode'; an unchecked checkbox for 'Hide source if a secondary translation exists'; an 'Editor link' text input with a description; a 'Special characters' text input with a description; and a 'Default dashboard view' section with radio buttons for 'Watched translations' (selected) and 'Suggested translations'. A 'Save' button is at the bottom left of the panel.

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

O menu tem estas opções:

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

filtrada por um dos seus idiomas primários.

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

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

Você pode configurar sua exibição de painel padrão pessoal na seção *Preferências* das configurações do perfil do usuário.

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

1.2.3 Perfil do usuário

O perfil do usuário é acessível clicando no ícone do usuário no topo direito do menu superior e, em seguida, no menu *Configurações*.

O perfil do usuário contém 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 você deseja traduzir clicando no botão para torná-los disponíveis também.


Idiomas

1.2.4 Idioma da interface





Escolha o idioma que você deseja exibir na UI.


Idiomas traduzidos

Escolha quais idiomas você prefere traduzir, e eles serão oferecidos na página principal de projetos assistidos, para que você tenha acesso mais fácil a essas todas as 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   MIT	76%	3	3		
 WeblateOrg/Django — Hungarian   GPL-3.0	69%	8	109	1	
 WeblateOrg/Django — Czech   GPL-3.0	96%	1	12	4	
 WeblateOrg/Django — Hebrew   GPL-3.0	92%	2	15		
 WeblateOrg/Djangojs — Czech   GPL-3.0	✓				
 WeblateOrg/Djangojs — Hebrew   GPL-3.0	✓				
 WeblateOrg/Djangojs — Hungarian   GPL-3.0	96%	2	6		
 WeblateOrg/Language names — Czech   GPL-3.0	✓				
 WeblateOrg/Language names — Hungarian   GPL-3.0	81%	4	5		
 WeblateOrg/Language names — Hebrew   GPL-3.0	✓				
 WeblateOrg/WeblateOrg — Czech   GPL-3.0	✓				
 WeblateOrg/WeblateOrg — Hungarian   GPL-3.0	✓				
 WeblateOrg/WeblateOrg — Hebrew   GPL-3.0	✓				

Powered by Weblate 4.6

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

Idiomas secundários

Você pode definir quais idiomas secundários são mostrados a você 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 displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main panel shows the 'Translation' view for a string. The source string is 'Files' in English, and the target string is 'קבצים' (Files) in Hebrew. The Czech target string is 'Soubory'. The interface includes navigation buttons, a 'Needs editing' checkbox, and action buttons 'Save', 'Suggest', and 'Skip'. Below the main panel, there are tabs for 'Nearby strings', 'Comments', 'Automatic suggestions', and 'Other languages'. A table shows the 'History' of translations for the string across different languages. On the right, there are sidebars for 'Glossary', 'String information', and 'Source string location'.

Language	Translated string
Hebrew	קבצים
Hungarian	Fájlok
English	Files

1.2.5 Preferências

Visão padrão do painel

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

Ver também:

[Lista de componentes](#)

Link do editor

Um link de código-fonte é mostrado no navegador web configurado nas *Configuração de componente* por padrão.

Dica: Ao definir o *Link do editor*, você usa o editor local para abrir o arquivo de código-fonte VCS de textos traduzidos. Você pode usar *Template markup*.

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

Ver também:

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

1.2.6 Notificações

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

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

Você 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 habilitado, você começará automaticamente a ver os projetos ao traduzir um texto. O valor padrão depende de `DEFAULT_AUTO_WATCH`.

Nota: Você não receberá notificações para suas próprias ações.

Web

late

Dashboard

Projects ▾

Languages ▾

Checks ▾

+

Your profile

Languages

Preferences

Notifications

Account

Profile

Licenses

Audit log

API access

Watched projects

☒ Automatically watch projects on contribution

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

Watched projects

Search...

Available:

WeblateOrg

Chosen:

WeblateOrg

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

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

Save

Notification settings

Other projects

Watched projects

Managed projects

Component wide notifications

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

Repository failure

Do not notify

Repository operation

Do not notify

Component locking

Do not notify

Changed license

Do not notify

Parse error

Do not notify

Comment on own translation

Instant notification

Mentioned in comment

Instant notification

New language

Do not notify

New translation component

Do not notify

New announcement

Instant notification

New alert

Do not notify

Translation notifications

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

New string

Do not notify

New contributor

Do not notify

New suggestion

Do not notify

New comment

Do not notify

Changed string

Do not notify

Translated string

Do not notify

Approved string

Do not notify

Pending suggestions

Do not notify

Strings needing action

Do not notify

Save

Powered by Weblate 4.6

About Weblate

Legal

Contact


Documentation

Donate to Weblate

1.2.7 Conta


A aba *Conta* permite configurar detalhes básicos da conta, conectar vários serviços que você pode usar para entrar no Weblate, remover completamente sua conta ou baixar seus dados de usuário (Veja [Exportação de dados de usuário 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 OAuth 2.0.


 Weblate

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

[+](#) [9](#) [...](#)

 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	<div>Change password</div>
 E-mail	weblate@example.org	<div>Disconnect</div>
 Google	weblate@example.org	<div>Disconnect</div>
 GitHub	123456	<div>Disconnect</div>
 Bitbucket	weblate	<div>Disconnect</div>

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

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

1.2.8 Perfil

Todos os campos desta página são opcionais e podem ser excluídos a qualquer momento e, ao preenchê-los, você está nos dando consentimento para compartilhar esses dados onde quer que seu perfil de usuário apareça.

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

1.2.9 Licenças

1.2.10 Acesso a API

Você pode obter ou redefinir seu token API de acesso aqui.

1.2.11 Registro de auditoria

O registro de auditoria acompanha as ações realizadas com sua conta. Ele registra endereço IP e navegador para cada ação importante com sua conta. As ações críticas também desencadeiam uma notificação para um endereço de e-mail principal.

Ver também:

Executando por trás de um proxy reverso

1.3 Traduzindo usando o Weblate

Obrigado pelo interesse em traduzir usando o Weblate. Os projetos podem ser configurados para tradução direta ou por meio de sugestões feitas por usuários 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 um número definido 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 qualquer um pode contribuir
- Visível apenas para um certo grupo de tradutores

Ver também:

Controle 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	0	

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

1.3.2 Links de tradução

Tendo navegado para um componente, um conjunto de links leva a sua tradução real. A tradução é ainda dividida em verificações individuais, como *Textos não traduzidos* ou *Textos que necessitam edição*. Se todo o projeto for traduzido, sem erro, *Todas os textos* ainda estão disponíveis. Alternativamente, você pode usar o campo de pesquisa para encontrar um texto ou termo específico.


Dashboard
Projects ▾
Languages ▾
Checks ▾

WebOrg / Django / Czech
translated 96%

Overview
Info
Search
Insights ▾
Files ▾
Tools ▾
Manage ▾
Share ▾
Watching ▾

Translation status

26 Strings 96%

183 Words 93%

Browse

Translate

Strings status

26 All strings — 183 words

Browse Edit Zen

25 Translated strings — 171 words

Browse Edit Zen

1 Strings needing action — 12 words

Browse Edit Zen

1 Not translated strings — 12 words

Browse Edit Zen

1 Strings needing action without suggestions — 12 words

Browse Edit Zen

3 Strings with any failing checks — 11 words

Browse Edit Zen

3 Translated strings with any failing checks — 11 words

Browse Edit Zen

1 Failed check: Unchanged translation — 4 words

Browse Edit Zen









1 Failed check: Mismatched full stop — 4 words

Browse Edit Zen

1 Failed check: Python format — 3 words

Browse Edit Zen

Other components

Component	Translated	Untranslated	Untranslated words	Checks	Suggestions	Comments
 Android  MIT	76%	3	3			
 Language names  GPL-3.0	<div></div>					
 Glossary WebOrg  GPL-3.0	<div></div>					
 Djangojs  GPL-3.0	<div></div>					

Browse all components

Powered by Weblate 4.6 [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.

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

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

1.3.4 Comentários

Três tipos de comentários podem ser publicados: para traduções, textos fonte ou para relatar erros de textos fonte quando esta funcionalidade é ativada utilizando *Habilitar revisões de fontes*. Escolha o adequado para o tópico que você deseja discutir. Os comentários de texto fonte são, em qualquer caso, bons para fornecer feedback sobre o texto original, por exemplo, que ele deve ser reformulado ou para fazer perguntas sobre ele.

Você pode usar a sintaxe do Markdown em todos os comentários e mencionar outros usuários usando @menção.

Ver também:

report-source, *Revisões de textos fonte*, *Habilitar revisões de fontes*

1.3.5 Variantes

As variantes são usadas para agrupar variantes de diferentes comprimentos do texto. O frontend de seu projeto pode, então, usar diferentes textos dependendo da tela ou do tamanho da janela.

Ver também:

variants, *Variantes*

1.3.6 Etiquetas

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

Ver também:

labels

1.3.7 Traduzindo

Na página de tradução, o texto fonte e uma área de edição para sua tradução são mostrados. Caso a tradução seja plural, são mostradas múltiplos textos 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 sendo usada). Campos de tradução para quaisquer idiomas secundários, que os tradutores venham a selecionar nas preferências, serão mostrados (veja *Idiomas secundários*) acima do texto fonte.

Embaixo da tradução, tradutores vão encontrar sugestões feita por outros, para serem aceitas (✓), aceitas com alterações (🔍) ou excluídas (🚫).

Plurais

Palavras mudando a forma para levar em conta sua designação numérica são chamadas de plurais. Cada idioma tem sua 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 suas regras para os plurais são diferentes.

O Weblate tem total suporte a cada uma dessas formas, em cada respectivo idioma (traduzindo cada plural separadamente). O número de campos e como ele é usado no aplicativo 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.

Ver também:

[Fórmula de plural](#)

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

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

translated 96%

[1/1](#)

[Custom search](#)

[Zen](#)

Position and priority

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

%(count)s slovo

Czech, Few

%(count)s slova

Czech, Other

%(count)s slov

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

☐ Needs editing

Save

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

16 seconds ago

Source string age

16 seconds ago

Translation file

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

Powered by Weblate 4.6

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

Atalhos de teclado

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

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

Atalho de teclado	Descrição
Alt+Home	Navega para a primeira tradução na pesquisa atual.
Alt+End	Navega para a última tradução na pesquisa atual.
Alt+PageUp ou Ctrl ↑ ou Alt ↑ ou Cmd ↑	Navega para a tradução anterior na pesquisa atual.
Alt+PageDown ou Ctrl+↓ ou Alt+↓ ou Cmd+↓	Navega para a próxima tradução na pesquisa atual.
Alt+Enter ou Ctrl+Enter ou Cmd+Enter	Salva a tradução atual.
Ctrl+Shift+Enter ou Cmd+Shift+Enter	Desmarca a tradução como precisando de edição e a envia.
Ctrl+E ou Cmd+E	Muda o foco para o editor de tradução.
Ctrl+U ou Cmd+U	Muda o foco para o editor de comentários.
Ctrl+M ou Cmd+M	Mostra a aba <i>Sugestões automáticas</i> , consulte <i>Sugestões automáticas</i> .
Ctrl+1 até Ctrl+9 ou Cmd+1 até Cmd+9	Copia objetos colocáveis de determinado número do texto fonte.
Ctrl+M+1 até 9 ou Cmd+M+1 até 9	Copia a tradução de máquina do número dado para a tradução atual.
Ctrl+I+1 até 9 ou Cmd+I+1 até 9	Ignora o item na lista de verificações com falha.
Ctrl+J ou Cmd+J	Mostra a aba de <i>Textos próximos</i> .
Ctrl+S ou Cmd+S	Muda o foco para o campo de pesquisa.
Ctrl+O ou Cmd+O	Copia o texto 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 usuário definidos em *Perfil do usuário*
- Caracteres por idioma fornecidos pelo Weblate (por exemplo, citações ou caracteres específicos RTL)
- Caracteres configurados usando *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 several panels: 'Glossary' (empty), 'String information' (showing context, explanation, labels, flags, source string location, string age, source string age, and translation file), and 'Screenshot context' (empty).

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

Contexto da tradução

Esta descrição contextual fornece informações relacionadas sobre o texto atual.

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

Capturas de tela Capturas de tela podem ser enviadas para o Weblate para melhor informar os tradutores sobre onde e como o texto é usado, veja [Contexto visual para textos](#).

Textos próximos Exibe mensagens próximas do arquivo de tradução. Estas também são geralmente usadas em um 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 aba mostra todos eles se forem considerados inconsistentes (veja [Inconsistente](#)). Você pode escolher qual

usar.

Memória de tradução Veja textos semelhantes traduzidos no passado, veja [Gerenciamento de memória](#).

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

Alterações recentes Lista de pessoas que mudaram esta mensagem recentemente usando Weblate.

Projeto Informações do projeto, como instruções para tradutores ou um diretório ou link para o texto no repositório do sistema de controle de versão que o projeto usa.

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

Histórico de tradução

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

Comprimento do texto traduzido

Weblate pode limitar o comprimento de uma tradução em várias formas para garantir o texto traduzido não é muito longo:

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

1.3.8 Sugestões automáticas

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

Ver também:

Você pode encontrar a lista de ferramentas suportadas em [Tradução de máquina](#).

1.3.9 Tradução automática

Você 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 você tenha selecionado um componente e um idioma:

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The 'Tools' menu is open, showing options like 'Search and replace', 'Bulk edit', 'Automatic translation' (which is highlighted), 'Data exports', and 'Failing checks'. The 'Automatic translation' settings panel is visible, showing the 'Automatic translation mode' set to 'Add as suggestion', the 'Search filter' set to 'Strings needing action', and the 'Automatic translation source' set to 'Machine translation'. Under 'Machine translation engines', 'Weblate' and 'Weblate Translation Memory' are listed in the 'Available' column, with 'Weblate' moved to the 'Chosen' column. A 'Score threshold' of 80 is set. An 'Apply' button is at the bottom of the panel.

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

Dois modos de operação são possíveis:

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

Você também pode escolher quais textos devem ser traduzidos automaticamente.

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

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

Ver também:

Mantendo 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, você fica bloqueado por um certo período até que você possa executar a operação novamente.

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

1.3.11 Pesquisar e substituir

Altere a terminologia efetivamente ou execute a correção em massa dos textos usando *Pesquisar e substituir* no menu *Ferramentas*.

Dica: Não se preocupe com bagunçar os textos. Este é um processo de dois passos que mostra uma pré-visualização dos textos editados antes que a alteração real seja confirmada.

1.3.12 Editor em massa

A edição em massa permite realizar uma operação sobre o número de textos. Você define os textos procurando por eles e configura algo a ser feito para corresponder a eles. As seguintes operações são suportadas:

- Alterar o estado de texto (por exemplo, para aprovar todos os textos não revisados).
- Ajustar os marcadores de tradução (veja [Personalizando comportamento usando marcadores](#))
- Ajustar as etiquetas de texto (veja labels)

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

Ver também:

[Extensão de edição em massa](#)

1.4 Baixando e enviando traduções

Você pode exportar arquivos a partir de uma tradução, fazer alterações e importá-los novamente. Isso permite trabalhar offline e, em seguida, mesclar mudanças de volta na tradução existente. Isso funciona mesmo que tenha sido alterado nesse meio tempo.

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

1.4.1 Baixando traduções

A partir do painel do projeto ou componente, arquivos traduzíveis podem ser baixados no menu *Arquivos*.

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

Você pode também baixar a tradução convertida em um dos formatos de localização amplamente utilizados. Os arquivos convertidos serão enriquecidos com dados fornecidos no Weblate, como contexto adicional, comentários ou marcadores. Vários formatos de arquivo estão disponíveis através do menu *Arquivos* ↓ *Personalizar o download*:

- Gettext PO
- XLIFF com extensões gettext
- XLIFF 1.1
- TermBase eXchange
- Translation Memory eXchange
- Gettext MO
- CSV
- Excel Open XML
- JSON
- Recurso de textos de Android
- Textos do iOS

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 / Czech' is visible, along with a 'translated 96%' indicator. The 'Files' menu is open, showing options: 'Download translation', 'Customize download' (highlighted), and 'Upload translation'. Below the menu, there's a table of 'Quick downloads' with three rows. The first row is 'File in original format as translated in the repository' with a link to 'gettext PO file'. The second and third rows are for 'All strings' and 'Strings needing action', both offering various file formats like CSV, gettext MO, XLIFF, TBX, TMX, and XLSX. Below this table is a 'Customize download' section with a dropdown for 'All strings', a 'File format' section with radio buttons for different formats (gettext PO is selected), and a 'Download' button.

Quick downloads	Download translation	Customize download	Upload translation
26 File in original format as translated in the repository			gettext PO file
26 All strings, converted files enriched with comments; suitable for offline translation	CSV	gettext MO	gettext PO
1 Strings needing action, converted files enriched with comments; suitable for offline translation	CSV	gettext MO	gettext PO

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 4.6 About Weblate Legal Contact Documentation Donate to Weblate

Ver também:

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

1.4.2 Enviando traduções

Quando você tiver feito suas alterações, use *Enviar tradução* no menu *Arquivos*.

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 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 'Download translation', 'Customize download', and 'Upload translation'. The 'Upload' form is displayed with the following sections:

- Upload**: A note stating 'The uploaded file will be merged with the current translation. To overwrite already translated strings, don't forget to turn it on.'
- File**: A 'Choose File' button and the text 'No file chosen'.
- File upload mode**: Four radio button options: 'Add as translation', 'Add as suggestion', 'Add as translation needing edit', and 'Replace existing translation file'.
- Processing of strings needing edit**: A dropdown menu currently set to 'Do not import'.
- Conflict handling**: A dropdown menu currently set to 'Update translated strings', with a note below: 'Whether to overwrite existing translations if the string is already translated.'
- Author name**: A text input field containing 'Weblate Test'.
- Author e-mail**: A text input field containing 'weblate@example.org'.
- Upload**: A dark blue button at the bottom of the form.

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

Formatos de arquivos suportados

Qualquer arquivo em um formato de arquivo suportado pode ser carregado, mas ainda é recomendado usar o mesmo formato de arquivo que o usado para tradução, caso contrário, alguns recursos podem não ser traduzidos corretamente.

Ver também:

Formatos de arquivos suportados

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

Métodos de importação

Estas são as opções apresentadas ao enviar arquivos de tradução:

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

Adicionar como sugestão (suggest**)** As traduções importadas são adicionadas como sugestões, faça isso quando quiser ter seus textos enviados revisados.

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

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

Atualizar textos fonte (source**)** Atualiza textos fonte em arquivo de tradução bilíngue. Isso é semelhante ao que *Atualizar arquivos PO para corresponder ao POT (msgmerge)* faz.

Esta opção é suportada apenas para alguns formatos de arquivo.

Adicionar novos textos (add**)** Adiciona novos textos à tradução. Pula a que já existe.

Caso você queira adicionar novos textos e atualizar traduções existentes, envie do arquivo pela segunda vez com *Adicionar como tradução*.

Esta opção está disponível apenas com *Gerenciar textos* ativado.

Ver também:

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

Gestão de conflitos

Define como lidar com textos enviados que já são traduzidos.

Textos necessitando de edição

Há também uma opção de como lidar com textos que necessitam de edição no arquivo importado. Tais textos podem ser manuseados de uma das três maneiras seguintes: “Não importar”, “Importar como texto que necessita edição” ou “Importar como traduzido”.

Substituindo autoria

Com permissões administrativas, você também pode especificar a autoria do arquivo enviado. Isso pode ser útil no caso de você ter recebido o arquivo 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 gerenciado sozinho, mas eles são armazenados juntos como um único componente que ajuda os administradores de projeto e tradutores multilíngues a manter alguma consistência entre os idiomas também. Os termos do glossário contendo palavras de texto atualmente traduzido são exibidos na barra lateral do editor de tradução.

1.5.1 Gerenciando glossários

Alterado na versão 4.5: Glossários são agora componentes de tradução comuns e você pode usar todos os recursos da Weblate neles — comentando, armazenando em um repositório remoto ou adicionando explicações.

Use qualquer componente como um glossário ligando *Usar como um glossário*. Você 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 utilizando *Compartilhar em projetos* do respectivo componente do glossário.

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

The screenshot displays the Weblate web interface for managing a glossary. The top navigation bar includes links to Dashboard, Projects, Languages, and Checks. The main header shows the project name 'WeblateOrg / Glossary' and the language 'Czech' with a 'translated 100%' indicator. The 'Overview' tab is selected, showing a summary of translation progress: 2 Strings (100%) and 3 Words (100%). Below this, the 'Strings status' section lists two entries: 'All strings — 3 words' and 'Translated strings — 3 words', each with 'Browse', 'Edit', and 'Zen' options. The 'Other components' table provides a detailed view of translation progress for different components, including 'Django' (96% translated, 12 untranslating words, 3 checks) and 'Language names' (100% translated, 1 check). A 'Browse all components' button is located at the bottom of the table.

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

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

Você 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 'WeblateOrg / Glossary WeblateOrg / Czech / Browse' is visible. A 'translated 100%' badge is in the top right. The main area displays a table with two columns: 'English' and 'Czech'. The first row shows 'machine translation' in English and 'strojový překlad' in Czech. The second row shows 'project' in English and 'projekt' in Czech. Navigation controls like '< < 1/1 > >' and 'All strings' are at the top. A '+ Add new glossary term' button is on the right. At the bottom, a footer contains links like 'Powered by Weblate 4.6', '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 os textos comuns. Você pode alternar recursos adicionais utilizando o menu *Ferramenta* para cada termo.

This screenshot provides a detailed view of the Weblate Glossary interface. The main panel shows a 'Glossary term' form with fields for 'English' (containing 'project') and 'Czech' (containing 'projekt'). There are buttons for 'Save', 'Suggest', and 'Skip'. Below the form, a 'Tools' dropdown menu is open, showing options: 'Delete string', 'Mark as read-only', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. To the right, a 'String information' sidebar shows 'String age' and 'Source string age' as '5 seconds ago'. At the bottom, a 'Nearby strings' section shows a list of terms, including 'machine translation' and 'project'. The footer is identical to the previous screenshot, showing 'Powered by Weblate 4.6' and various links.

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.

Ver também:

Personalizando comportamento usando 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.

Ver também:

Personalizando comportamento usando 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.

Ver também:

Personalizando comportamento usando marcadores

Variantes

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

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

Ver também:

variants

1.6 Verificações e correções

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

Uma vez que enviar uma tradução com uma verificação de falha, isso é imediatamente mostrado ao usuário:

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

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

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

1/1

Custom search

'%(count)s word'

Position

Zen

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

Czech, Few

několik slov

Czech, Other

%(count)s slov

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

☐ Needs editing

Save

Suggest

Skip

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

21 seconds ago

Source string age

22 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 4.6 [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 alguns erros comuns em textos traduzidos automaticamente. Use isso com cuidado para não causar erros por meio disto.

Ver também:

AUTOFIX_LIST

1.6.2 Verificações de qualidade

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

Ver também:

CHECK_LIST, *Personalizando comportamento usando marcadores*

1.6.3 Verificações de tradução

Executada a cada alteração na tradução, ajudando os tradutores a manter traduções de boa qualidade.

Marcação de 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 sejam encontrados na tradução.

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

Há palavras duplicadas de forma consecutiva

O texto contém a mesma palavra duas vezes seguidas:

Novo na versão 4.1.

Verifica se não ocorrem palavras duplicadas consecutivas em uma 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 seguir glossário

Novo na versão 4.5.

A tradução não segue os termos definidos em um glossário.

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

- Ele faz a correspondência exata dos textos, espera-se que o glossário contenha termos em todas as variantes.
- A verificação de cada texto 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.

Ver também:

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

Espaço duplo

A tradução contém espaço duplo

Verifica se o espaço duplo está presente na tradução para evitar falsos positivos em outras verificações relacionadas ao espaço.

A verificação é falsa quando o espaço duplo é encontrado no texto fonte, o que significa que o espaço duplo é intencional.

Textos formatados

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

Weblate tem suporte a verificar textos de formato em vários idiomas. A verificação não é ativada automaticamente, somente se um texto for sinalizado adequadamente (por exemplo, “c-format” para formato C). O Gettext adiciona isso automaticamente, mas você provavelmente terá que adicioná-lo manualmente para outros formatos de arquivo ou se seus arquivos PO não forem gerados por **xgettext**.

Isso pode ser feito por unidade (ver [Informações adicionais sobre textos fonte](#)) na [Configuração de componente](#). Tê-lo definido por componente é mais simples, mas pode levar a falsos positivos no caso de o texto não ser interpretado como um texto 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, você pode usar [Espaços reservados](#) genéricos.

Além de verificar, isso também destacará os textos e formatação para inseri-los facilmente em textos traduzidos:

Weblate
 Dashboard Projects Languages Checks

Django / Czech / Translate
 translated 96%

[1/1](#)

Custom search

'%(count)s word'

Position and priority

Zen

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

%(count)s slovo

Czech, Few

%(count)s slova

Czech, Other

%(count)s slov

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

☐ Needs editing

Save Suggest Skip

Glossary

English Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

16 seconds ago

Source string age

16 seconds ago

Translation file

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

Nearby strings 20 Comments Automatic suggestions Other languages 3 History

No matching activity found.

Browse all component changes

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

Texto de interpolação AngularJS

O texto de interpolação AngularJS não corresponde à fonte

Texto de formato nomeado	Seu saldo é {{amount}} {{ currency }}
Sinalizador para habilitar	<i>angularjs-format</i>

Ver também:

Interpolação de texto do AngularJS

Formato C

O texto de formato C não corresponde à fonte

Texto de formato simples	Há %d maçãs
Texto de formato de posição	Seu saldo é %1\$d %2\$s
Sinalizador para habilitar	<i>c-format</i>

Ver também:

C format strings, C printf format

Formato C#

O texto de formato C# não corresponde à fonte

Texto de formato de posição	Há {0} maçãs
Sinalizador para habilitar	<i>c-sharp-format</i>

Ver também:

Formato de string do C#

Literais de modelo de ECMAScript

Os literais de modelo de ECMAScript não correspondem à fonte

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

Ver também:

Template strings

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
Sinalizador para habilitar	<i>i18next-interpolation</i>

Ver também:

Interpolação i18next

Formato Java

O texto de formato java não corresponde à fonte

Texto de formato simples	Há %d maçãs
Texto de formato de posição	Seu saldo é %1\$d %2\$s
Sinalizador para habilitar	<i>java-format</i>

Ver também:

[Java Format Strings](#)

MessageFormat do Java

O texto de MessageFormat do Java não corresponde à fonte

Texto de formato de posição	Há {0} maçãs
Sinalizador para habilitar	<i>java-messageformat</i> habilita a verificação incondicionalmente
	<i>auto-java-messageformat</i> habilita verificação somente se houver um texto de formato na fonte

Ver também:

[Java MessageFormat](#)

Formato JavaScript

O texto de formato JavaScript não corresponde à fonte

Texto de formato simples	Há %d maçãs
Sinalizador para habilitar	<i>javascript-format</i>

Ver também:

[JavaScript formatting strings](#)

Formato Lua

O formato de texto Lua não corresponde com a fonte

Texto de formato simples	Há %d maçãs
Sinalizador para habilitar	<i>lua-format</i>

Ver também:

[Lua formatting strings](#)

Espaços reservados de porcentagem

Os espaços reservados de porcentagem não correspondem à fonte

Novo na versão 4.0.

Texto de formato simples	Há %number% maçãs
Sinalizador para habilitar	<i>percent-placeholders</i>

Formato Perl

O texto de formato Perl não corresponde à fonte

Texto de formato simples	Há %d maçãs
Texto de formato de posição	Seu saldo é %1\$d %2\$s
Sinalizador para habilitar	<i>perl-format</i>

Ver também:

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

Formato PHP

O texto de formato PHP não corresponde à fonte

Texto de formato simples	Há %d maçãs
Texto de formato de posição	Seu saldo é %1\$d %2\$s
Sinalizador para habilitar	<i>php-format</i>

Ver também:

[Documentação de PHP sprintf](#), [PHP Format Strings](#)

Formato de chaves Python

O texto de formato de chaves Python não corresponde à fonte

Texto de formato simples	Há {} maçãs
Texto de formato nomeado	Seu saldo é {amount} {currency}
Sinalizador para habilitar	<i>python-brace-format</i>

Ver também:

[Formato de chaves Python](#), [Python Format Strings](#)

Formato Python

O texto de formato Python não corresponde à fonte

Texto de formato simples	Há %d maçãs
Texto de formato nomeado	Seu saldo é %(amount) %(currency)
Sinalizador para habilitar	<i>python-format</i>

Ver também:

[Formatação de texto Python, Python Format Strings](#)

Formato Qt

O texto de formato Qt não corresponde à fonte

Texto de formato de posição	Há %1 maçãs
Sinalizador para habilitar	<i>qt-format</i>

Ver também:

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

Formato de plural Qt

O texto de formato de plural do Qt não corresponde à fonte

Texto de formato de plural	Há %Ln maçã (s)
Sinalizador para habilitar	<i>qt-plural-format</i>

Ver também:

[Guia de i18n do Qt](#)

Formato Ruby

O texto de formato Ruby não corresponde à fonte

Texto de formato simples	Há %d maçãs
Texto de formato de posição	Seu saldo é %1\$f %2\$s
Texto de formato nomeado	Seu saldo é %+.2<amount>f %<currency>s
Texto do modelo nomeado	Seu saldo é %{amount} %{currency}
Sinalizador para habilitar	<i>ruby-format</i>

Ver também:

[Ruby Kernel#sprintf](#)

Formato Scheme

O formato do texto Scheme não corresponde com a fonte

Texto de formato simples	Há ~d maçãs
Sinalizador para habilitar	<i>scheme-format</i>

Ver também:

Srfi 28, formato de Scheme Chicken, Saída formatada de Guile Scheme

Formatação Vue I18n

A formatação Vue I18n não corresponde com a fonte

Formatação nomeada	Há {count} maçãs
Formatação Rails i18n	Há %{count} maçãs
Mensagens da localidade vinculada	@:message.dio @:message.the_world!
Sinalizador para habilitar	<i>vue-format</i>

Ver também:

Formatação Vue I18n, Mensagens de localidade Vue I18n

Foi traduzido

Este texto foi traduzido no passado

Significa que um texto já foi traduzido. Isso pode acontecer quando as traduções foram revertidas no VCS ou perdidas de outra forma.

Inconsistente

Este texto tem mais de uma tradução neste projeto ou não é traduzida em alguns componentes.

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

A verificação falha em diferentes traduções de um texto dentro de um projeto. Isso também pode levar a inconsistências nas verificações exibidas. Você pode encontrar outras traduções deste texto na aba *Outras ocorrências*.

Nota: Esta verificação também é disparada no caso de o texto estar traduzido em um componente e não em outro. Ela pode ser usado como uma maneira rápida de manusear manualmente textos que não estão traduzidos em alguns componentes apenas clicando no botão *Usar esta tradução* exibido em cada linha na aba *Outras ocorrências*.

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

Ver também:

Mantendo traduções iguais entre componentes

Letra Kashida usada

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.

Ver também:

[Kashida na Wikipédia](#)

Links Markdown

Os links Markdown não correspondem com a fonte

Novo na versão 3.5.

Os links Markdown não correspondem com a fonte.

Ver também:

[Links Markdown](#)

Referências Markdown

As referências de link Markdown não correspondem à fonte

Novo na versão 3.5.

As referências de link Markdown não correspondem à fonte.

Ver também:

[Links Markdown](#)

Sintaxe Markdown

A sintaxe de Markdown não corresponde à fonte

Novo na versão 3.5.

A sintaxe de Markdown não corresponde à fonte

Ver também:

[Elementos span Markdown](#)

Comprimento 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 o número de gráficos, 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 o texto.

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 número máximo de linhas. Caso o número de linhas não seja fornecido, é considerado um texto de uma linha.

Você também pode configurar a fonte utilizada pelas diretivas `font-*` (veja [Personalizando comportamento usando 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: Você pode querer definir as diretivas `font-*` na [Configuração de componente](#) para ter a mesma fonte configurada para todas as strings dentro de um componente. Você pode sobrescrever esses valores por texto, caso você precise personalizá-la por texto.

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

Ver também:

[Gerenciando fontes](#), [Personalizando comportamento usando marcadores](#), [Comprimento máximo da tradução](#)

\n não correspondente

O número de \n nas traduções não corresponde ao fonte

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

Caractere de dois pontos não correspondente

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).

Ver 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.

Ver 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).

Ver 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).

Ver também:

[Ponto final na Wikipédia](#)

Ponto de interrogação não correspondente

Fonte e tradução não terminam com uma 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).

Ver 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 arquivos desktop.

Ver também:

[Ponto e vírgula na Wikipédia](#)

Quebras de linha descasadas

Número 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 o número 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 um texto fonte foram traduzidas. Especificações sobre como cada forma plural é usada podem ser encontradas na definição do texto.

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: Você pode usar uma expressão regular como espaço reservado.

A tradução não contém alguns espaços reservados. Estes são extraídos do arquivo de tradução ou definidos manualmente usando o sinalizador `placeholders`, outros podem ser separados com ponto e vírgula, textos com o espaço podem ser colocados entre aspas:

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

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

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

Ver também:

Personalizando comportamento usando marcadores

Espaçamento de 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.

Ver 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 arquivo de tradução ou definida manualmente usando 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 nos textos fonte por uma boa razão, de forma que omissões ou adições podem levar a problemas de formatação quando o texto traduzido é colocado em uso.

Ver também:

Linha em branco no final

Espaços no início

Fonte e tradução não iniciam com o mesmo número de espaços

Um espaço no início de um texto é normalmente usado para recuo na interface e, portanto, importante de manter.

Linha em branco no final

Fonte e tradução não terminam com uma nova linha

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

Ver 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 não alterada

A fonte e a tradução são idênticas

Acontece se o texto fonte e a tradução correspondente forem idênticos, até pelo menos uma das formas de plural. Alguns textos normalmente encontrados em todos os idiomas são ignorados, e várias marcações são ignoradas. Isto reduz o número de falsos positivos.

Esta verificação pode ajudar a encontrar textos erroneamente não traduzidos.

O comportamento padrão desta verificação é excluir palavras da lista negra embutida da verificação. Estas são palavras que frequentemente não estão sendo traduzidas. Isto é útil para evitar falsos positivos em textos curtos, que consistem apenas em uma única palavra que é a mesma em vários idiomas. Esta lista negra pode ser desabilitada adicionando `strict-same` ao texto ou componente.

Ver também:

Configuração de componente, Personalizando comportamento usando 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 habilitada utilizando o sinalizador `safe-html` (veja [Personalizando comportamento usando marcadores](#)). Há também o autocorretor acompanhado que pode automaticamente corrigir a marcação.

Ver 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 contendo uma URL. Nesse caso, a tradução tem de ser uma URL válido.

Marcação 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.

Espaço com largura zero

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.

Ver também:

[Espaço de largura zero na Wikipédia](#)

1.6.4 Verificações de texto fonte

Estas verificações podem ajudar os desenvolvedores a melhorar a qualidade dos textos fonte.

Reticências

O texto usa três pontos (...) ao invés do caractere de reticências (...)

Isto falha quando o texto 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*).

Ver também:

[Reticências na Wikipédia](#)

Não traduzido a muito tempo

Este texto não foi traduzido há muito tempo

Novo na versão 4.1.

Quando o texto não é traduzida há muito tempo, pode indicar um problema em um texto fonte, tornando difícil a tradução.

Várias verificações com falha

As traduções em diversos idiomas têm verificações com falha

Várias traduções deste texto têm falhas nas verificações de qualidade. Isto é normalmente uma indicação de que algo poderia ser feito para melhorar o texto 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 que seria melhor corrigi-la no texto fonte.

Várias variáveis sem nome

Há várias variáveis sem nome no texto, o que impossibilita aos tradutores reordená-las

Novo na versão 4.1.

Há várias variáveis sem nome no texto, o que impossibilita aos tradutores reordená-las.

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

Não pluralizado

O texto é usado como plural, mas não está usando formas de plural

O texto é usado como um plural, mas não usa formas de plural. Caso o seu sistema de tradução tenha suporte a isso, você 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 usando operações booleanas, parênteses ou pesquisas específicas de campo podem ser usadas para encontrar os textos que você deseja.

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

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

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

At the bottom of the search area, there is a 'Search' button.

Powered by Weblate 4.6 [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 cordas que contêm qualquer uma delas são mostradas. Para procurar uma frase exata, coloque “a frase de pesquisa” entre aspas (ambas aspas simples (') e duplas (") funcionarão: "este é um texto entre aspas" ou 'outro texto entre aspas'.

1.7.2 Campos

source:TEXTO Pesquisa em texto fonte sem diferenciar maiúscula de minúscula.

target:TEXTO Pesquisa em texto destino sem diferenciar maiúscula de minúscula.

context:TEXTO Pesquisa em texto de contexto sem diferenciar maiúscula de minúscula.

key:TEXTO Pesquisa em texto chave sem diferenciar maiúscula de minúscula.

note:TEXTO Pesquisa em texto de comentário sem diferenciar maiúscula de minúscula.

location:TEXTO Pesquisa em texto de localização sem diferenciar maiúscula de minúscula.

priority:NÚMERO Prioridade do texto.

added:DATA E HORA Data e hora de quando o texto foi adicionado ao Weblate.

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

pending:BOOLEANO Texto 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 texto (pending, translated, untranslated).

language:TEXTO Idioma de destino do texto.

component:TEXTO URL amigável do componente, veja *URL amigável do componente*.

project:TEXTO Slug do projeto, veja *URL amigável*.

changed_by:TEXTO O texto foi alterado pelo autor com o nome de usuário fornecido.

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

change_time:DATA E HORA Texto foi alterado na data, suporta *Operações de campo*; ao contrário de changed, isso inclui eventos que não alteram o conteúdo e você pode aplicar uma filtragem de ação personalizada utilizando 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 O texto tem verificação com falha.

dismissed_check:TEXTO O texto tem verificação dispensada.

comment:TEXTO Pesquisa nos comentários de usuários.

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

Você pode combinar pesquisas usando 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

Você 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

Você pode fazer uma consulta de correspondência exata em diferentes campos de texto usando o operador `=`. Por exemplo, para procurar por todos os textos fonte que correspondem exatamente a `hello world`, use: `source:="hello world"`. Para pesquisar por expressões de uma única palavra, você não precisa de aspas. Por exemplo, para procurar por todos os textos fonte que correspondam a `hello`, você pode utilizar: `source:=hello`.

1.7.6 Pesquisando por alterações

Novo na versão 4.4.

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

Por exemplo, uma pesquisa por textos marcados 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, você também pode especificar uma expressão regular como `r "regexp"`.

Por exemplo, para pesquisar por todos os textos 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:

Weblate
Dashboard
Projects
Languages
Checks

WeblateOrg / Django / Czech / Translate
translated 96%

1/1

Custom search
'%(count)s word'

Zen

Translation

English

Singular

'%(count)s word

Plural

'%(count)s words

Czech, One

'%(count)s slovo

Czech, Few

'%(count)s slova

Czech, Other

'%(count)s slov

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

☐ Needs editing

Save
Suggest
Skip

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

16 seconds ago

Source string age

16 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

1.7.9 Ordenando os resultados

Há muitas opções para ordenar os textos de acordo com as suas necessidades:

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 / Czech / Translate' is visible, along with a 'translated 96%' indicator. The main area shows a string being translated from English to Czech. The English string is 'The string uses three dots (...) instead of an ellipsis character (...)'. The Czech string is empty. A dropdown menu is open, showing sorting options: 'Position and priority', 'Position', 'Priority', 'Labels', 'Source string', 'Translated string', 'Age of string', 'Number of words', 'Number of comments', 'Number of failing checks', and 'Key'. Below the string input, there are buttons for 'Save', 'Suggest', and 'Skip'. At the bottom, there's a 'New comment' section with a text area and a 'Save' button. On the right, there's a sidebar with 'Nearby strings', 'Comments', 'Automatic suggestions', and 'Other languages'. The 'Comments' tab is active, showing a list of comments with fields for '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 aproxima seus usuários de você, aproximando você de seus tradutores. Depende de você decidir quantos recursos deseja usar.

A lista a seguir não é uma lista completa de maneiras de configurar o Weblate. Você 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 usuário* significa um usuário que tenha acesso à tradução. Pode ser qualquer usuário autenticado se o projeto for público, ou um usuário que tenha uma permissão *Traduzir* para o projeto.

1.8.2 Estados de tradução

Cada texto traduzida pode estar em um dos seguintes estados:

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

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

Aguardando revisão A tradução está feita, mas não revisada. Ela é armazenada no arquivo como uma tradução válida.

Aprovado A tradução foi aprovada na revisão. Não pode mais ser alterada por tradutores, mas apenas por revisores. Tradutores só podem adicionar sugestões a ela.

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

Os estados são representados nos arquivos de tradução quando possível.

Dica: In case file format you use does not support storing states, you might want to use *Marcar traduções não alteradas como “Necessita edição”* addon to flag unchanged strings as needing editing.

Ver também:

Capacidades dos tipos de tradução, Fluxos de trabalho de tradução

1.8.3 Tradução direta

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

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

Configuração	Valor	Nota
Habilitar revisões	desativada	Configurada a nível de projeto.
Habilitar sugestões	ativada	É útil para os usuários serem capazes de sugerir quando não têm certeza.
Votação de sugestões	desativada	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	<i>Usuários</i>	Or <i>Translate with per-project access control</i> .
Grupo de revisores	N/D	Não usada.

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 aceita como tradução.

- *Qualquer usuário* pode adicionar sugestões.
- *Qualquer usuário* pode votar em sugestões.
- Sugestões tornam-se traduções quando dado um número predeterminado de votos.

Configuração	Valor	Nota
Habilitar revisões	desativada	Configurada a nível de projeto.
Habilitar sugestões	ativada	
Votação de sugestões	desativada	
Aceitar sugestões automaticamente	1	Você pode definir um valor mais alto para exibir mais revisões por pares.
Grupo de tradutores	<i>Usuários</i>	Or <i>Translate with per-project access control</i> .
Grupo de revisores	N/D	Não usada, 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.

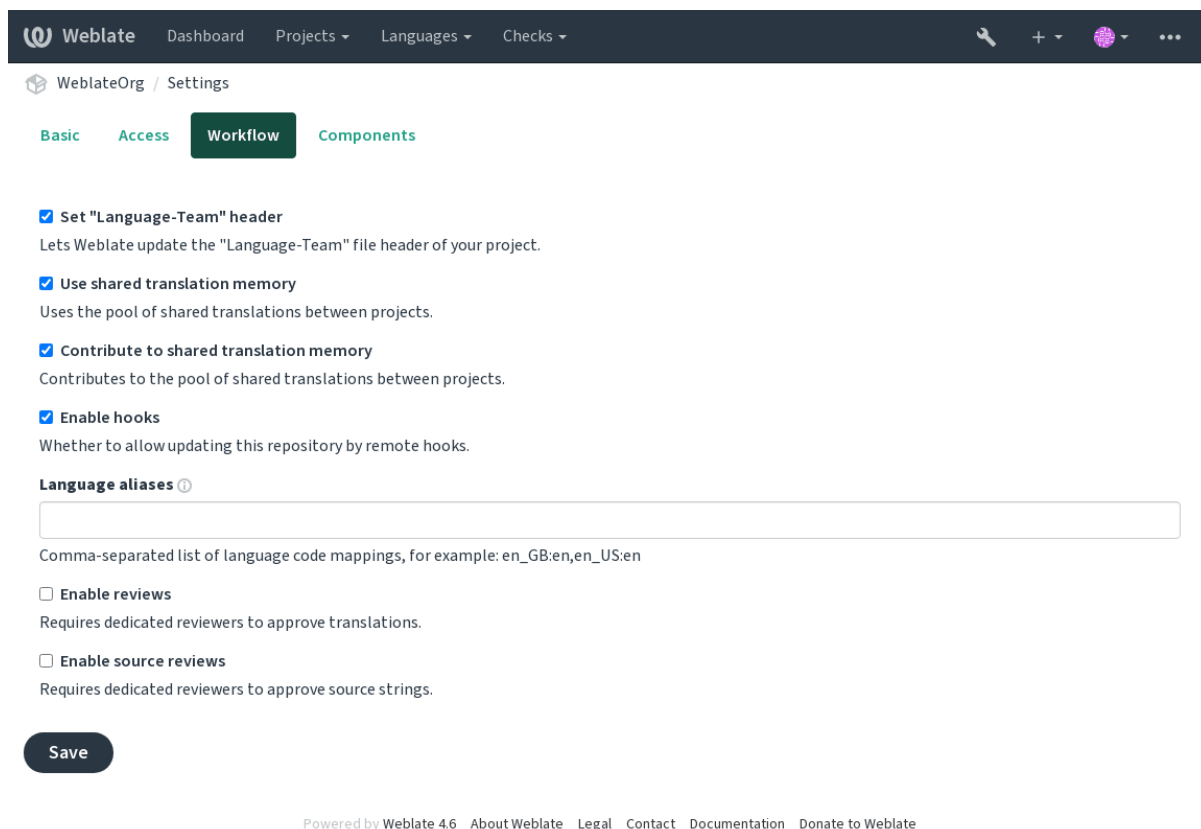
Com revisores dedicados você tem dois grupos de usuários, um capaz de enviar traduções e outro capaz de revisá-los para garantir que as traduções sejam consistentes e que a qualidade seja boa.

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

Configuração	Valor	Nota
Habilitar revisões	ativada	Configurada a nível de projeto.
Habilitar sugestões	desativada	É útil para os usuários serem capazes de sugerir quando não têm certeza.
Votação de sugestões	desativada	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	<i>Usuários</i>	Or <i>Translate</i> with <i>per-project access control</i> .
Grupo de revisores	<i>Revisores</i>	Or <i>Review</i> with <i>per-project access control</i> .

1.8.6 Ativando 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 *Gerenciar* → *Configurações*):

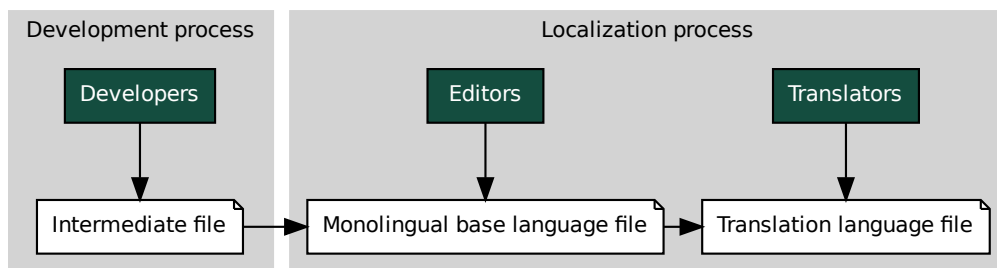


Nota: Dependendo da configuração do Weblate, a configuração pode não estar disponível para você. Por exemplo, no Hosted Weblate, isso não está disponível para projetos hospedados gratuitamente.

1.8.7 Rota de qualidade para os textos fonte

Em muitos casos, os textos fonte do idioma fonte vêm de desenvolvedores, porque eles escrevem o código e fornecem textos iniciais. No entanto, os desenvolvedores muitas vezes não são falantes nativos do idioma fonte e não fornecem qualidade desejada dos textos fonte. A tradução intermediária pode ajudá-lo a lidar com isso - há uma rota de qualidade adicional para os textos entre desenvolvedores e tradutores e usuários.

Ao definir um *Arquivo de idioma intermediário*, este arquivo será usado como fonte para os textos, mas será editado para o idioma fonte para polí-lo. Uma vez que o texto esteja pronto no idioma fonte, ele também estará disponível para os tradutores traduzirem em idiomas adicionais.



Ver também:

Arquivo de idioma intermediário, Arquivo de idioma da base monolíngue, Formatos bilíngues e monolíngues

1.8.8 Revisões de textos fonte

Com o *Habilitar revisões de fontes* ativado, o processo de revisão pode ser aplicado em textos fonte. Uma vez ativado, os usuários podem relatar problemas nos textos fonte. O processo real depende se você usa formatos bilíngues ou monolíngues.

Para formatos monolíngues, a revisão de texto fonte se comporta da mesma forma que com *Revisores dedicados* - uma vez que o problema é relatado no texto fonte, ele é marcado como *Necessita edição*.

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

Ver 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 você der acesso de push ao seu repositório, as traduções podem acontecer sem interação, a menos que algum conflito de mesclagem ocorra.

1. Configure seu repositório Git para informar Weblate quando houver alguma alteração, veja *Ganchos 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 *Push ao fazer commit* on your *Configuração de componente* in Weblate, this will make Weblate push changes to your repository whenever they happen at Weblate.

Ver também:

Localização contínua, Evitando 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 arquivo de tradução é alterado tanto na Weblate como no repositório upstream ao mesmo tempo. Você geralmente pode evitar isso mesclando traduções Weblate antes de fazer alterações nos arquivos de tradução (por exemplo, antes de executar o msgmerge). Apenas diga ao Weblate para fazer o commit de todas as traduções pendentes (você pode fazer isso em *Manutenção do repositório* no menu *Gerenciar*) e mesclar o repositório (se o push automático não estiver ligado).

Se você já encontrou um conflito de mesclagem, a maneira mais fácil é resolver todos os conflitos localmente na sua estação de trabalho - é simplesmente adicionar Weblate como um repositório remoto, mesclá-lo com upstream e corrigir quaisquer conflitos. Uma vez que você fizer push das mudanças de volta, Weblate será capaz de usar a versão unida sem nenhuma outra ação especial.

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 você estiver usando vários ramos na Weblate, você 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

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

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

No caso de arquivos 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, você 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
```

Ver também:

Como exportar o repositório Git que a Weblate usa?, Localização contínua, Evitando conflitos de mesclagem, Weblate Client

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, você pode ter que mesclar esses ramos (dependendo do seu fluxo de trabalho de desenvolvimento) descartando as diferenças:

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

Ver também:

Mantendo traduções iguais entre componentes

Como traduzir projetos multiplataforma?

Weblate tem suporte a uma grande variedade de formatos de arquivo (veja *Formatos de arquivos suportados*) e a abordagem mais fácil é utilizar o formato nativo para cada plataforma.

Uma vez que você tenha adicionado todos os arquivos de tradução da plataforma como componentes em um projeto (veja *Adicionando projetos e componentes de tradução*), você pode utilizar o recurso de propagação de tradução (ligado por padrão, e pode ser desligado na *Configuração de componente*) para traduzir textos para todas as plataformas ao mesmo tempo.

Ver também:

Mantendo 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 você tem acesso SSH a esta máquina, você pode usar o repositório diretamente.

Para acesso anônimo, você pode querer executar um servidor Git e deixá-lo servir o repositório para o mundo exterior. Alternativamente, você 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?*).
- Você 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, eliminando compromissos de mesclagem), mescla alterações e diz ao Weblate para redefinir o conteúdo no repositório upstream.

Claro que você é livre para misturar tudo isso como quiser.

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

Você 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 seus arquivos 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. Você pode atualizar o repositório principal com traduções do Weblate executando:

```
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 você pode ver na interface administrativa, basta seguir o link *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 “committer”, configurado quando você cria um componente de tradução. Você pode alterá-lo na administração a qualquer momento.

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

Ver também:

Configuração de componente

1.9.2 Uso

Como faço para rever as traduções dos outros?

- Há vários fluxos de trabalho baseados em revisões disponíveis em Weblate, veja *Fluxos de trabalho de tradução*.
- Você 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 você pode optar por navegar pelas traduções feitas por outros desde uma data determinada.

Ver também:

Fluxos de trabalho de tradução

Como posso fornecer feedback sobre um texto fonte?

Nas abas de contexto abaixo da tradução, você pode utilizar a aba *Comentários* para fornecer feedback sobre um texto fonte, ou discuti-lo com outros tradutores.

Ver 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.
- Você pode importar tradução arquivos de memória de tradução existente 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 usando um compêndio ou uma base de dados de tradução similar.
- Você pode configurar *tmserver* com todos os bancos de dados que você tem e deixar o Weblate utilizá-lo. Isto é bom quando você quer utilizá-lo várias vezes durante a tradução.
- Outra opção é traduzir todos os projetos relacionados em uma única instância Weblate, o que o fará pegar automaticamente as traduções de outros projetos também.

Ver também:

Tradução de máquina, Sugestões automáticas, Memória de tradução

O Weblate atualiza arquivos de tradução além de traduções?

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

Ver 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.

Você é livre para definir seus próprios idiomas na interface administrativa, você só precisa fornecer informações sobre isso.

Ver também:

Definições de idioma

Weblate pode destacar alterações em um texto aproximado?

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

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

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

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

Por que é que o Weblate ainda mostra os textos de tradução antigos quando eu atualizei o modelo?

Weblate não tenta manipular os arquivos de tradução de nenhuma outra forma que não seja permitir que os tradutores traduzam. Por isso, também não atualiza os arquivos traduzíveis quando o modelo ou código-fonte tiverem sido alterados. Você 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 arquivos de tradução, pois de outra forma geralmente acabará com alguns conflitos para mesclar.

Por exemplo, com os arquivos PO gettext, você pode atualizar os arquivos de tradução utilizando a ferramenta **msgmerge**:

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

Caso você queira fazer a atualização automaticamente, você pode instalar a extensão *Atualizar arquivos PO para corresponder ao POT (msgmerge)*.

Ver também:

updating-target-files

1.9.3 Soluções de problemas

As solicitações às vezes falham com o erro “muitos arquivos abertos”

Isto acontece às vezes quando o seu repositório Git cresce demais e você 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
```

Ver 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 você quer acessar no seu Weblate. Por exemplo:

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

Ver também:

Configuração de hosts permitidos

O que significa “Existem mais arquivos para o idioma único (en)”?

Isso normalmente acontece quando você tem arquivo de tradução para idioma fonte. Weblate mantém o controle de textos fonte e reserva o idioma fonte para isso. O arquivo adicional para o mesmo idioma não é processado.

- Caso a tradução para o idioma fonte seja desejada, altere o *Idioma fonte* nas configurações do projeto.
- Caso o arquivo de tradução para o idioma fonte não seja necessário, remova-o do repositório.
- Caso o arquivo de tradução para o idioma fonte seja necessário, mas deva ser ignorado pelo Weblate, ajuste o *Filtro de idioma* para excluí-lo.

Dica: Você também pode receber mensagens de erro semelhantes para outros idiomas. Nesse caso, o motivo mais provável é que vários arquivos 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 (f r e f r _ F R). Veja *language-parsing-codes* para mais detalhes.

1.9.4 Recursos

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.

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

Weblate also supports VCS-less operation, see *Arquivos 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.

Ver também:

Integração com controle de versão

Como o Weblate credita os tradutores?

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

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

Ver também:

[`list_translators`](#), [`../devel/reporting`](#)

Por que o Weblate força mostrar todos os arquivos PO em uma única árvore?

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

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

Por que o Weblate usa códigos de idioma como `sr_Latn` ou `zh_Hant`?

Estes são códigos de idioma definidos por [RFC 4646](#) para melhor indicar que eles são idiomas realmente diferentes, em vez de modificadores utilizados anteriormente de forma errada (para variantes `@latin`) ou códigos de país (para chinês).

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`.

Ver também:

[*Definições de idioma*](#)

1.10 Formatos de arquivos suportados

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

Ver também:

[Translation Related File Formats](#)

Nota: Ao escolher um formato de arquivo para o seu aplicativo, é melhor colocar algum formato bem estabelecido no kit de ferramentas/plataforma que você usa. Dessa forma, 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 em um único arquivo e tradução (exemplos típicos são [GNU gettext](#), [XLIFF](#) ou [Textos de Apple iOS](#)). Por outro lado, formatos monolíngues identificam o texto por ID, e cada arquivo de idioma contém apenas o mapeamento desses para qualquer idioma (tipicamente [Recurso de textos de Android](#)). Alguns formatos de arquivo são utilizados em ambas as variantes, veja a descrição detalhada abaixo.

Para o uso correto de arquivos monolíngues, Weblate requer acesso a um arquivo contendo uma lista completa de textos para traduzir com sua fonte – este arquivo é chamado de [Arquivo de idioma da base monolíngue](#) dentro do Weblate, embora a nomenclatura possa variar no seu paradigma.

Além disso este fluxo de trabalho pode ser estendido usando *Arquivo de idioma intermediário* para incluir textos fornecidos pelos desenvolvedores, mas não para ser usado como está nos textos finais.

1.10.2 Detecção automática

O Weblate pode detectar automaticamente vários formatos de arquivos generalizados, mas essa detecção pode prejudicar seu desempenho e limitará recursos específicos para o formato de arquivo 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 ⁹	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>Propri- edades GWT</i>	mono	sim	sim	não	não	não	
<i>Tradu- ções 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 textos de Android</i>	mono	sim	sim ⁷	não	não	sim ⁷	
<i>Textos de Apple iOS</i>	bilíngue	não	sim	não	não	não	
<i>Textos do PHP</i>	mono	não ¹¹	sim	não	não	não	
<i>Arquivos JSON</i>	mono	não	não	não	não	não	
<i>Arquivos i18next do JSON</i>	mono	sim	não	não	não	não	
<i>Arquivos JSON go-i18n</i>	mono	sim	não	não	não	não	
<i>Arquivo ARB</i>	mono	sim	sim	não	não	não	
<i>JSON We- bExtension</i>	mono	sim	sim	não	não	não	

continua na próxima página

Tabela 1 – continuação da página anterior

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>Arquivos de recursos .XML</i>	mono	não	sim	não	não	sim [?]	
<i>Arquivos CSV</i>	ambos	não	sim	sim	sim	não	necessita edição
<i>Arquivos YAML</i>	mono	não	sim	não	não	não	
<i>Arqui- vos Ruby YAML</i>	mono	sim	sim	não	não	não	
<i>Arquivos DTD</i>	mono	não	não	não	não	não	
<i>Arqui- vos XML simples</i>	mono	não	não	não	não	sim [?]	
<i>Arquivos RC do Windows</i>	mono	não	sim	não	não	não	
<i>Excel Open XML</i>	mono	não	sim	sim	sim	não	necessita edição
<i>Arquivos de meta- dados de loja de aplicativos</i>	mono	não	não	não	não	não	
<i>Arquivos de legenda</i>	mono	não	não	não	sim	não	
<i>Arquivos HTML</i>	mono	não	não	não	não	não	
<i>Formato OpenDo- cument</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 de Inno Setup INI</i>	mono	não	não	não	não	não	
<i>Formato TermBase eXchange</i>	bilíngue	não	sim	não	não	sim [?]	
<i>Arquivos texto</i>	mono	não	não	não	não	não	

Textos somente leitura

Novo na versão 3.10.

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

1.10.4 GNU gettext

O formato mais usado para traduzir software livre.

As informações contextuais armazenadas no arquivo são suportadas ajustando seus cabeçalhos ou vinculando-se a arquivos fonte correspondentes.

O arquivo de PO de texto bilíngue normalmente se parece com este:

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

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

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

Configuração de componente típica do Weblate	
Máscara do arquivo	<code>po/*.*.po</code>
Arquivo de idioma da base monolíngue	<i>Vazio</i>
Modelo para novas traduções	<code>po/messages.pot</code>
Formato de arquivo	<i>Arquivo Gettext PO</i>

Ver também:

[devel/gettext](#), [devel/sphinx](#), [Gettext no Wikipédia](#), [PO Files](#), *Atualizar variável ALL_LINGUAS no arquivo “configure”*, *Personalizar saída do gettext*, *Atualizar arquivo LINGUAS*, *Gerar arquivos MO*, *Atualizar arquivos PO para corresponder ao POT (msgmerge)*

¹ Veja *Formatos bilíngues e monolíngues*

² Os plurais são necessários para localizar adequadamente os textos com contagem variável.

³ Os comentários podem ser usados para passar informações adicionais sobre o texto para traduzir.

⁴ O contexto é usado para diferenciar textos idênticos usados em diferentes escopos (por exemplo, *Sun* pode ser usado como um nome abreviado do dia “Sunday” ou como o nome de nossa estrela mais próxima).

⁵ A localização de um texto no código-fonte pode ajudar tradutores proficientes a descobrir como o texto é usado.

⁶ Veja *Personalizando comportamento usando marcadores*

⁷ Estados adicionais suportados pelo formato do arquivo, 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 *Especificando marcadores de tradução*.

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

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

Gettext monolíngue

Alguns projetos decidem usar o gettext como formatos monolíngues – eles codificam apenas os IDs em seu código-fonte e o texto então precisa ser traduzido para todos os idiomas, incluindo o inglês. Isso é suportado, embora você tenha que escolher este formato de arquivo explicitamente ao importar componentes para o Weblate.

O arquivo 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 arquivo 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 do arquivo	po/* .po
Arquivo de idioma da base monolíngue	po/en .po
Modelo para novas traduções	po/messages .pot
Formato de arquivo	<i>Arquivo Gettext PO (monolíngue)</i>

1.10.5 XLIFF

Formato baseado em XML criado para padronizar arquivos 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.

Ver 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 arquivo é parcialmente processado e mapeado para o estado “Necessita edição” no Weblate (os seguintes estados são utilizados para marcar o texto como necessitando editar se houver um alvo presente: `new`, `needs-translation`, `needs-adaptation`, `needs-l10n`). Caso o atributo `state` esteja faltando, um texto é considerado traduzido assim que um elemento `<target>` existir.

Se o texto de tradução tiver `approved="yes"`, ela também será importada para a Weblate como “Aprovado”, qualquer outra coisa será importada como “Aguardando revisão” (que corresponde à especificação XLIFF).

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

- O atributo `state` só é adicionado no caso de texto ser marcado como necessitando de edição.
- O atributo `approved` só é adicionado no caso de texto ter sido revisto.
- 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, a fim de ver e alterar o estado aprovado de textos.

Da mesma forma, ao importar tais arquivos (no formulário de envio), você deve escolher *Importador como traduzido* em *Processando os textos que necessitam edição*.

Ver 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 você quiser mantê-los, você tem que adicionar o marcador `xml:space="preserve"` ao texto.

Por exemplo:

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

Especificando marcadores de tradução

Você pode especificar marcadores de tradução adicionais (veja *Personalizando comportamento usando marcadores*) utilizando 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 `fonte` 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 *Gerenciando fontes*).

Chaves de textos

Weblate identifica as unidades no arquivo 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, tornando-a mais adequada para a exibição de Weblate em vez de `id`. O `resname` tem de ser único em todo o arquivo 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</i> típica do Weblate para XLIFF bilíngue	
Máscara do arquivo	<code>localizations/*.xliff</code>
Arquivo de idioma da base monolíngue	<i>Vazio</i>
Modelo para novas traduções	<code>localizations/en-US.xliff</code>
Formato de arquivo	<i>Arquivo de tradução XLIFF</i>

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

Ver 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 do aplicativo.

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

Ver também:

[Java properties na Wikipédia](#), [Mozilla and Java properties files](#), [Formatar o arquivo de propriedades Java](#), [Limpar arquivos de tradução](#)

1.10.7 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 do arquivo	src/app/Bundle_*.properties
Arquivo de idioma da base monolíngue	src/app/Bundle.properties
Modelo para novas traduções	Vazio
Formato de arquivo	Propriedades GWT

Ver também:

[Guia de localização GWT](#), [Tutorial de internacionalização GWT](#), [Mozilla and Java properties files](#), [Formatar o arquivo de propriedades Java](#), [Limpar arquivos de tradução](#)

1.10.8 Traduções de INI

Novo na versão 4.1.

Formato de arquivo INI para traduções.

As traduções de INI são geralmente usadas como traduções monolíngues.

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

Nota: O Weblate só extrai chaves de seções dentro de um arquivo INI. Caso seu arquivo INI não tenha seções, você pode querer usar [Traduções para Joomla](#) ou [Propriedades Java](#).

Ver também:

[INI Files](#), [Propriedades Java](#), [Traduções para Joomla](#), [Traduções de Inno Setup INI](#)

1.10.9 Traduções de Inno Setup INI

Novo na versão 4.1.

Formato de arquivo Inno Setup INI para traduções.

As traduções de Inno Setup INI são geralmente 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 do arquivo	language/*.isl
Arquivo de idioma da base monolíngue	language/en.isl
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo Inno Setup INI

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

Ver também:

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

1.10.10 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 do arquivo	language/*/com_foobar.ini
Arquivo de idioma da base monolíngue	language/en-GB/com_foobar.ini
Modelo para novas traduções	Vazio
Formato de arquivo	<i>Arquivo de idioma Joomla</i>

Ver também:

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

1.10.11 .ts de Qt Linguist

Formato de tradução usado em aplicativos baseados em Qt.

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

<i>Configuração de componente</i> típica do Weblate ao usar como bilíngue	
Máscara do arquivo	i18n/app.*.ts
Arquivo de idioma da base monolíngue	Vazio
Modelo para novas traduções	i18n/app.de.ts
Formato de arquivo	<i>Arquivo de tradução Qt Linguist</i>

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

Ver também:

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

1.10.12 Recurso de textos de Android

Formato de arquivo específico do Android para traduzir aplicativos.

Os recursos de texto do Android são monolíngues, o *Arquivo de idioma da base monolíngue* é armazenado em um local diferente dos outros `res/values/strings.xml`.

<i>Configuração de componente</i> típica do Weblate	
Máscara do arquivo	<code>res/values-*/strings.xml</code>
Arquivo de idioma da base monolíngue	<code>res/values/strings.xml</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de arquivo	<i>Recurso de textos de Android</i>

Ver também:

Documentação de recursos de textos de Android, [Android string resources](#)

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

```
<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 em um arquivo diferente e não ser disponibilizado para tradução.

Este script pode ajudar a preprocessar seus arquivos `strings.xml` e traduções existentes: <https://gist.github.com/paour/11291062>

1.10.13 Textos de Apple iOS

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

Os textos de Apple iOS geralmente são usados como traduções bilíngues.

<i>Configuração de componente</i> típica do Weblate	
Máscara do arquivo	<code>Resources/*.lproj/Localizable.strings</code>
Arquivo de idioma da base monolíngue	<code>Resources/en.lproj/Localizable.strings</code> ou <code>Resources/Base.lproj/Localizable.strings</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de arquivo	<i>Textos do iOS (UTF-8)</i>

Ver também:

Documentação de “arquivos de textos” da Apple, [Mac OSX strings](#)

1.10.14 Textos do PHP

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

Exemplo de arquivo:

```
<?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 do arquivo	lang/*/texts.php
Arquivo de idioma da base monolíngue	lang/en/texts.php
Modelo para novas traduções	lang/en/texts.php
Formato de arquivo	<i>Textos do PHP</i>

Textos do PHP de Laravel

Alterado na versão 4.1.

Os arquivos 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',
];
```

Ver também:

[PHP, Localização de Laravel](#)

1.10.15 Arquivos 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 arquivos JSON de estrutura aninhada também são suportados.

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

O formato JSON é usado principalmente para traduzir aplicativos implementados no JavaScript.

O Weblate atualmente suporta várias variantes das traduções JSON:

- Arquivos simples de chave/valor, usados por exemplo por *vue-i18n* ou *react-intl*.
- Arquivos com chaves aninhadas.
- *Arquivos i18next do JSON*
- *Arquivos JSON go-i18n*
- *JSON WebExtension*
- *Arquivo ARB*

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

Exemplo de arquivo:

```
{
  "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 arquivos aninhados também são suportados (veja acima para os requisitos), tal arquivo pode parecer:

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

Dica: O arquivo JSON e Arquivo de estrutura JSON aninhada podem ambos lidar com o mesmo tipo de arquivos. Ambos preservam a estrutura JSON existente ao traduzir.

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

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

Configuração de componente típica do Weblate

Máscara do arquivo	<code>langs/translation-*.json</code>
Arquivo de idioma da base monolíngue	<code>langs/translation-en.json</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de arquivo	<i>Arquivo de estrutura JSON aninhada</i>

Ver também:

[JSON](#), [Personalizar saída JSON](#), [Limpar arquivos de tradução](#),

1.10.16 Arquivos i18next do JSON

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

[i18next](#) é uma estrutura de internacionalização escrita dentro e para JavaScript. O Weblate suporta seus arquivos de localização com recursos como plurais.

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

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

Exemplo de arquivo:

```
{
  "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 do arquivo	langs/*.json
Arquivo de idioma da base monolíngue	langs/en.json
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo JSON i18next

Ver também:

JSON, JSON Format de i18next, *Personalizar saída JSON*, *Limpar arquivos de tradução*

1.10.17 Arquivos JSON go-i18n

Novo na versão 4.1.

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

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

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

Ver também:

JSON, go-i18n, *Personalizar saída JSON*, *Limpar arquivos de tradução*,

1.10.18 Arquivo ARB

Novo na versão 4.1.

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

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

Ver também:

[JSON](#), [Application Resource Bundle Specification](#), [Internationalizing Flutter apps](#), [Personalizar saída JSON](#), [Limpar arquivos de tradução](#)

1.10.19 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 arquivo usado ao traduzir extensões para Mozilla Firefox ou Google Chromium.

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

Exemplo de arquivo:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara do arquivo	<code>_locales/*/messages.json</code>
Arquivo de idioma da base monolíngue	<code>_locales/en/messages.json</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de arquivo	<i>Arquivo JSON WebExtension</i>

Ver também:

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

1.10.20 Arquivos de recursos .XML

Novo na versão 2.3.

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

<i>Configuração de componente</i> típica do Weblate	
Máscara do arquivo	Resources/Language.*.resx
Arquivo de idioma da base monolíngue	Resources/Language.resx
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo de recurso .NET

Ver também:

.NET Resource files (.resx), *Limpar arquivos de tradução*

1.10.21 Arquivos CSV

Novo na versão 2.4.

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

- Arquivos 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 *Arquivo CSV* como um formato de arquivo.
- Arquivos com dois campos – fonte e tradução (nesta ordem). Escolha *Arquivo CSV simples* como um formato de arquivo.
- 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 *Arquivo de idioma da base monolíngue* quando seus arquivos são monolíngues (veja *Formatos bilíngues e monolíngues*).

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

Exemplo de arquivo:

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

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

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

Ver também:

[CSV](#)

1.10.22 Arquivos YAML

Novo na versão 2.9.

Os arquivos YAML simples com chaves de textos e valores. Weblate também extrai textos de listas ou dicionários.

Exemplo de um arquivo YAML:

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

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

Ver também:

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

1.10.23 Arquivos Ruby YAML

Novo na versão 2.9.

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

Exemplo de arquivo YAML de i18n do Ruby:

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

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

Ver também:

[YAML](#), [Arquivos YAML](#)

1.10.24 Arquivos DTD

Novo na versão 2.18.

Exemplo de arquivo DTD:

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

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

Ver também:

[Mozilla DTD format](#)

1.10.25 Arquivos XML simples

Novo na versão 3.9.

Exemplo de um arquivo 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 do arquivo	locale/*.xml
Arquivo de idioma da base monolíngue	locale/en.xml
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo XML simples

Ver também:

[Flat XML](#)

1.10.26 Arquivos RC do Windows

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

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

Exemplo de arquivo RC do Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

STRINGTABLE
BEGIN
    IDS_MSG1                "Hello, world!\n"
```

(continua na próxima página)

(continuação da página anterior)

IDS_MSG2	"Orangutan has %d banana.\n"
IDS_MSG3	"Try Weblate at http://demo.weblate.org/!\n"
IDS_MSG4	"Thank you for using Weblate."
END	

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

Ver também:[Windows RC files](#)

1.10.27 Arquivos de metadados de loja de aplicativos

Novo na versão 3.5.

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

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

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

<i>Configuração de componente</i> típica do Weblate	
Máscara do arquivo	fastlane/android/metadata/*
Arquivo de idioma da base monolíngue	fastlane/android/metadata/en-US
Modelo para novas traduções	fastlane/android/metadata/en-US
Formato de arquivo	<i>Arquivos de metadados de loja de aplicativos</i>

Dica: Caso você não queira traduzir certos textos (por exemplo, changelogs), marque-as como somente leitura (veja [Personalizando comportamento usando marcadores](#)). Isto pode ser automatizado por [Editor em massa](#).

1.10.28 Arquivos de legenda

Novo na versão 3.7.

Weblate pode traduzir vários arquivos de legenda:

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

<i>Configuração de componente</i> típica do Weblate	
Máscara do arquivo	path/*.srt
Arquivo de idioma da base monolíngue	path/en.srt
Modelo para novas traduções	path/en.srt
Formato de arquivo	<i>Arquivo de legenda SubRip</i>

Ver também:[Subtitles](#)

1.10.29 Excel Open XML

Novo na versão 3.2.

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

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

1.10.30 Arquivos 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 arquivos HTML e oferecido para a tradução.

Ver também:[HTML](#)

1.10.31 Arquivos 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:

- Arquivo texto simples
- Arquivo texto DokuWiki
- Arquivo texto MediaWiki

Ver também:[Simple Text Documents](#)

1.10.32 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 arquivos OpenDocument e oferecido para a tradução.

Ver também:

[OpenDocument Format](#)

1.10.33 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 arquivos Adobe InDesign Markup Language e oferecido para a tradução.

1.10.34 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 do arquivo	<code>tbx/*.*tbx</code>
Arquivo de idioma da base monolíngue	<i>Vazio</i>
Modelo para novas traduções	<i>Vazio</i>
Formato de arquivo	<i>Arquivo TermBase eXchange</i>

Ver também:

[TBX na Wikipédia](#), [TBX](#), [Glossário](#)

1.10.35 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](#).

Ver também:

[Translation Related File Formats](#)

1.11 Integração com controle de versão

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 você deseja usar tem que ser acessível ao Weblate. Com um repositório disponível publicamente, você 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 usuário dedicado para fazer push registrado no GitHub, Bitbucket, Codeberg e GitLab (com o nome de usuário *weblate*, e-mail `hosted@weblate.org` e chamado *Weblate push user*). Você precisa adicionar esse usuário 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 usuário *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 usuário *weblate*, você pode configurar o *Repositório do código-fonte* e a *URL de push do repositório* utilizando o protocolo SSH (por exemplo, `git@github.com:WeblateOrg/weblate.git`).

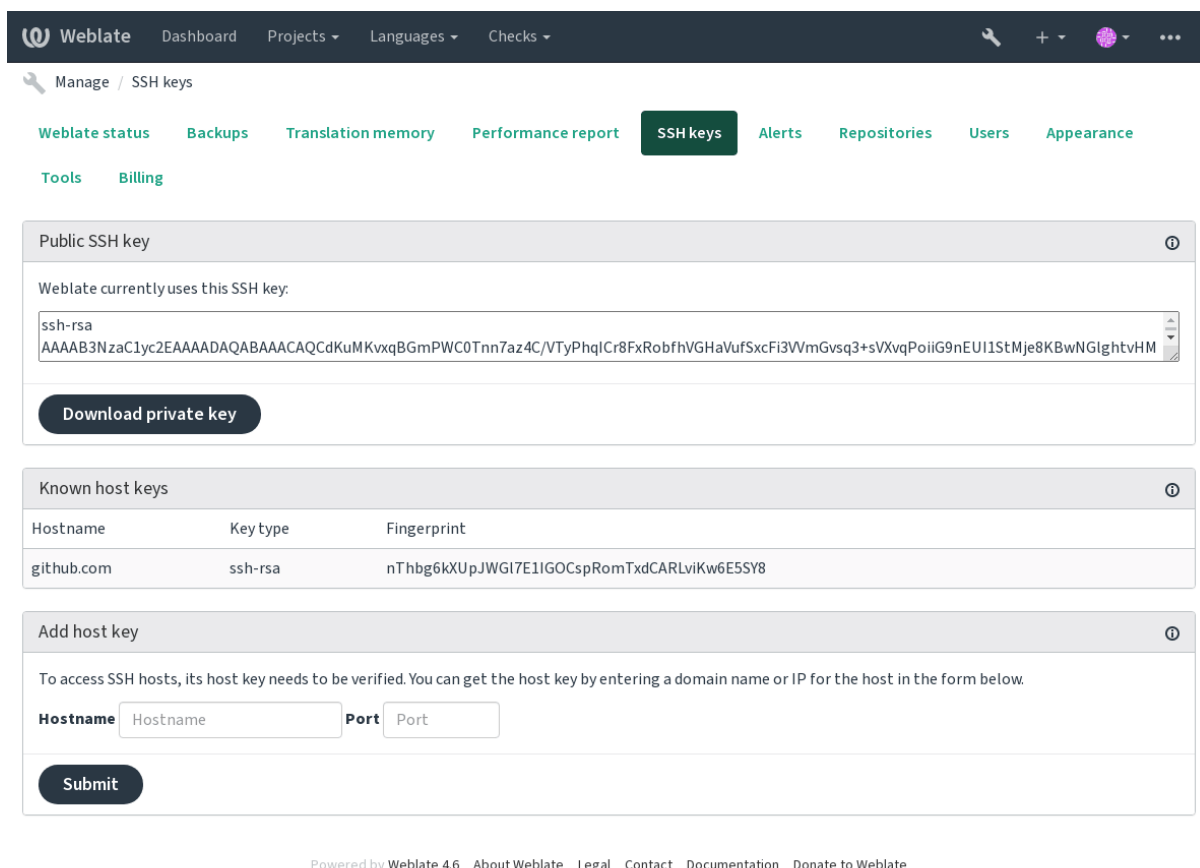
Repositórios SSH

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 Weblate web interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a secondary navigation bar with links to Manage / SSH keys, Weblate status, Backups, Translation memory, Performance report, SSH keys (highlighted), Alerts, Repositories, Users, and Appearance. Under SSH keys, there are links for Tools and Billing.

The main content area is divided into three sections:

- Public SSH key:** A box titled "Public SSH key" with an information icon. It states "Weblate currently uses this SSH key:" and displays a long SSH key in a monospace font. Below the key is a button labeled "Download private key".
- Known host keys:** A table titled "Known host keys" with an information icon. It has three columns: Hostname, Key type, and Fingerprint. The table contains one entry for github.com with key type ssh-rsa and fingerprint nThbg6kXUpJWG17E1GOCspRomTxdCARLviKw6E5SY8.
- Add host key:** A box titled "Add host key" with an information icon. It contains a message: "To access SSH hosts, its host key needs to be verified. You can get the host key by entering a domain name or IP for the host in the form below." Below this is a form with two input fields: "Hostname" and "Port". A "Submit" button is at the bottom.

At the bottom of the page, there is a footer with the text "Powered by Weblate 4.6" and links to About Weblate, Legal, Contact, Documentation, and Donate to Weblate.

Chave SSH do Weblate

A chave pública do Weblate está visível para todos os usuários 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 senha 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 delas para uso posterior.

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

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

Manage / SSH keys

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 currently uses this SSH key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCDkUMKvxqBGmPWC0Tnn7az4C/VTyPhqICr8FxRobfhVGHavufSxcFi3VvmGvsq3+sVXvqPoiIG9nEU11StMje8KBwNGightvHM
```

[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](#)

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

Repositórios do GitHub

O acesso via SSH é possível (veja [Repositórios SSH](#)), mas caso você precise acessar mais de um repositório, você 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 sua conta GitHub, veja [Criando um token de acesso para uso em linha de comando](#).

Para configurações maiores, geralmente é melhor criar um usuário 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 você deseja traduzir. Essa abordagem também é usada para o Hosted Weblate, há usuário dedicado *weblate* para isso.

Ver também:

[Acessando repositórios do Hosted Weblate](#)

URLs internas do Weblate

Compartilhe uma configuração de repositório entre diferentes componentes, fazendo 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. Você 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 compartilhando um repositório; por exemplo, [Squash de commits git](#).

Repositórios HTTPS

Para acessar repositórios HTTPS protegidos, inclua o nome de usuário e a senha na URL. Não se preocupe, o Weblate irá remover essas informações quando a URL for mostrada aos usuários (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 seu nome de usuário ou senha contiver caracteres especiais, eles devem ser codificados para URL; por exemplo, `https://usuario%40example.com:%24senha%23@bitbucket.org/...`

Usando proxy

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

Isto pode ser feito utilizando as variáveis de ambiente `http_proxy`, `https_proxy` e `all_proxy` (como descrito na documentação do [cURL](#)) ou aplicando-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 usuário executando Weblate (veja também [Permissões do sistema de arquivos](#)) e com `HOME=$DATA_DIR/home` (veja [DATA_DIR](#)), caso contrário o Git executado pelo Weblate não o utilizará.

Ver também:

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

1.11.2 Git

Ver 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 em 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 usuário precisa ser feita em `DATA_DIR/home/.git`.

Auxiliares de remotos do Git

Você 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 coloque em algum lugar em 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 usando Bazaar:

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

Para o repositório `hello` de `selenic.com` usando 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* utilizando 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.

Ver 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.

Você precisa configurar credenciais de API para fazer isso funcionar.

Ver 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* usando 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, sendo 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.

Ver 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.

Você precisa configurar credenciais de API para fazer isso funcionar.

Ver 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* usando 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, sendo 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.

Ver também:

Fazendo push das alterações do Weblate

Fazendo push de alterações para o Pagure como merge requests

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

Você precisa configurar credenciais de API para fazer isso funcionar.

Ver também:

PAGURE_USERNAME, PAGURE_TOKEN, PAGURE_CREDENTIALS

1.11.6 Gerrit

Novo na versão 2.2.

Adiciona uma camada fina sobre o *Git* usando 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 você 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.

Ver 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, permitindo que os usuários mantenham um clone completo do repositório interno e façam commit localmente.

Nota: O Weblate tenta detectar 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 você 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 usando o layout padrão eram suportados.

Credenciais de Subversion

Weblate espera que você tenha aceito o certificado com antecedência (e suas credenciais, se necessário). Ele procurará inseri-las no diretório `:set:DATA_DIR`. Aceite o certificado utilizando `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
```

Ver também:

`DATA_DIR`

1.11.9 Arquivos locais

Novo na versão 3.8.

O Weblate também pode operar sem um VCS remoto. As traduções iniciais são importadas carregando-as. Mais tarde, você pode substituir arquivos individuais enviando arquivos ou adicionando textos de tradução diretamente do Weblate (atualmente disponível apenas para traduções monolíngues).

No fundo, o Weblate cria um repositório de Git para você e todas as alterações são rastreadas. Caso você decida mais tarde usar um VCS para armazenar as traduções, você já tem um repositório dentro do Weblate pode basear 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](#). Você pode utilizá-la diretamente ou por [Weblate Client](#).

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 você pode obter em seu perfil. Use-o no cabeçalho `Authorization`:

ANY /

O comportamento de solicitação genérico para a API, os cabeçalhos, códigos de status 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` – token opcional para se autenticar

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*) – verbose description of the result (for HTTP status codes other than 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 usando API
- **web_url** (*string*) – URL para acessar esse recurso usando navegador web

Códigos de status

- 200 OK – quando a solicitação foi corretamente tratada
- 201 Created – when a new object was created successfully
- 204 No Content – when an object was created successfully
- 400 Bad Request – quando faltam parâmetros de 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 usuários anônimos e 5000 solicitações por hora para usuários 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 status 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	Número de segundos até que a janela de “ratelimit” seja reiniciada

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

Ver 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 Usuários

Novo na versão 4.0.

GET /api/users/

Retorna uma lista de usuários se você tiver permissões para ver usuários gerenciados. Se não, então você pode ver apenas seus próprios detalhes.

Ver também:

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

POST /api/users/

Cria um novo usuário.

Parâmetros

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

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

Retorna informações sobre os usuários.

Parâmetros

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

Objeto JSON de resposta

- **username** (*string*) – nome de usuário de um usuário

- **full_name** (*string*) – nome completo de um usuário
- **email** (*string*) – e-mail de um usuário
- **is_superuser** (*boolean*) – se o usuário é um superusuário
- **is_active** (*boolean*) – se o usuário está ativo
- **date_joined** (*string*) – data de criação do usuário
- **groups** (*array*) – link 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 usuário.

Parâmetros

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

Objeto JSON de resposta

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

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

Altera os parâmetros do usuário.

Parâmetros

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

Objeto JSON de resposta

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

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

Exclui todas as informações do usuário e marca o usuário como inativo.

Parâmetros

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

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

Associa grupos a um usuário.

Parâmetros

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

Parâmetros de formulário

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

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

Lista as estatísticas de um usuário.

Parâmetros

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

Objeto JSON de resposta

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

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

Lista assinaturas de um usuário.

Parâmetros

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

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

Associa assinaturas com um usuário.

Parâmetros

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

Objeto JSON da solicitação

- **notification** (*string*) – Nome da notificação registrada
- **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 usuário.

Parâmetros

- **username** (*string*) – Nome de usuário do usuário
- **subscription_id** (*int*) – ID da notificação registrada

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

int: subscription_id / Edita uma assinatura associada a um usuário.

Parâmetros

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

- **subscription_id** (*int*) – ID da notificação registrada

Objeto JSON da solicitação

- **notification** (*string*) – Nome da notificação registrada
- **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 usuário.

Parâmetros

- **username** (*string*) – Nome de usuário do usuário
- **subscription_id** (*int*) – ID da notificação registrada

Objeto JSON da solicitação

- **notification** (*string*) – Nome da notificação registrada
- **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* / Exclui uma assinatura associada a um usuário.

Parâmetros

- **username** (*string*) – Nome de usuário do usuário
- **subscription_id** – Nome da notificação registrada
- **subscription_id** – int

1.12.4 Grupos

Novo na versão 4.0.

GET /api/groups/

Retorna uma lista de grupos se você tiver permissões para ver grupos gerenciados. Se não, então você pode ver apenas os grupos dos quais o usuário faz parte.

Ver 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*) /

Retorna informações 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*) – link para funções associadas; veja `GET /api/roles/(int:id)/`
- **projects** (*array*) – link para projetos associados; veja `GET /api/projects/(string:project)/`
- **components** (*array*) – link para componentes associados; veja `GET /api/components/(string:project)/(string:component)/`
- **componentlist** (*array*) – link 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)/`

Exclui o grupo.

Parâmetros

- `id(int)` – ID do grupo

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

Associa 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` Exclui o 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` Exclui o projeto de um grupo.

Parâmetros

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

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

Associa idiomas 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` Exclui idioma 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* Exclui 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 usuário. Se o usuário for superusuário, então a lista de todas as funções existentes será retornada.

Ver 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"
  ],
  "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*) /

Exclui a função.

Parâmetros

- **id** (*int*) – ID da função

1.12.6 Idiomas

GET /api/languages/

Retorna uma lista de todos os idiomas.

Ver também:

Os atributos do objeto de idioma estão documentados em [GET /api/languages/ \(string:language\) /](#).

POST /api/languages/

Cria um novo idioma.

Parâmetros

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

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

Retorna informações sobre um idioma.

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 do idioma.

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 do idioma.

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) /`
Exclui o idioma.

Parâmetros

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

GET `/api/languages/(string: language)/statistics/`
Retorna estatísticas para um idioma.

Parâmetros

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

Objeto JSON de resposta

- **total** (*int*) – número total de textos

- **total_words** (*int*) – número total de palavras
- **last_change** (*timestamp*) – alterações mais recentes no idioma
- **recent_changes** (*int*) – número total de alterações
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **translated_words** (*int*) – número de palavras traduzidas
- **translated_words_percent** (*int*) – porcentagem de palavras traduzidas
- **translated_chars** (*int*) – número de caracteres traduzidos
- **translated_chars_percent** (*int*) – porcentagem de caracteres traduzidos
- **total_chars** (*int*) – número total de caracteres
- **fuzzy** (*int*) – número de textos aproximados (marcados para edição)
- **fuzzy_percent** (*int*) – porcentagem de textos aproximados (marcados para edição)
- **failing** (*int*) – número de textos com falha
- **failing** – porcentagem de textos com falha

1.12.7 Projetos

GET /api/projects/

Retorna uma lista de todos os projetos.

Ver 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 do projeto

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

Retorna informações sobre um projeto.

Parâmetros

- **project** (*string*) – URL amigável 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 status 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*) – *Habilitar revisões*
- **source_review** (*boolean*) – *Habilitar revisões de fontes*
- **set_language_team** (*boolean*) – *Definir o cabeçalho do “Language-Team”*
- **enable_hooks** (*boolean*) – *Habilitar ganchos*
- **instructions** (*string*) – *Instruções de tradução*
- **language_aliases** (*string*) – *Aliases de 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 amigável do projeto
- **component** (*string*) – URL amigável 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 amigável do projeto

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

Novo na versão 3.9.

Exclui um projeto.

Parâmetros

- **project** (*string*) – URL amigável 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 aceitando os mesmos parâmetros.

Parâmetros

- **project** (*string*) – URL amigável 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 status do repositório VCS. Este contém apenas um resumo geral de todos os repositórios para o projeto. Para obter um status mais detalhado, use `http: get:/api/components/(string:project)/(string:component)/repository/`.

Parâmetros

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

Objeto JSON da solicitação

- **operation** (*string*) – Operação para realizar: uma de push, pull, commit, re-set, 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 amigável 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 [Arquivos 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 amigável do projeto

Parâmetros de formulário

- **file zipfile** – Arquivo ZIP para carregar no 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 arquivos usando 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
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
```

(continua na próxima página)

(continuação da página anterior)

```

"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 em um projeto.

Novo na versão 3.8.

Parâmetros

- **project** (*string*) – URL amigável 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*) – número total de textos
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas
- **words_percent** (*float*) – porcentagem de palavras traduzidas

GET /api/projects/ (string: *project*) /statistics/
 Retorna estatísticas para um projeto.

Novo na versão 3.8.

Parâmetros

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

Objeto JSON de resposta

- **total** (*int*) – número total de textos
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **total_words** (*int*) – número total de palavras

- **translated_words** (*int*) – número de palavras traduzidas
- **words_percent** (*float*) – porcentagem de palavras traduzidas

1.12.8 Componentes

GET `/api/components/`

Retorna uma lista de componentes de tradução.

Ver 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 amigável do projeto
- **component** (*string*) – URL amigável 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 controle de versão
- **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 arquivo
- **template** (*string*) – Arquivo de idioma da base monolíngue
- **edit_template** (*string*) – Editar o arquivo base
- **intermediate** (*string*) – Arquivo de idioma intermediário
- **new_base** (*string*) – Modelo para novas traduções
- **file_format** (*string*) – Formato de arquivo
- **license** (*string*) – Licença da tradução
- **agreement** (*string*) – Acordo de colaborador
- **new_lang** (*string*) – Adicionando 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 push do repositório
- **check_flags** (*string*) – Marcadores de tradução
- **priority** (*string*) – Prioridade
- **enforced_checks** (*string*) – Verificações forçadas

- **restricted** (*string*) – *Acesso restrito*
- **repoweb** (*string*) – *Navegador do repositório*
- **report_source_bugs** (*string*) – *Endereço do relatório de erros do texto fonte*
- **merge_style** (*string*) – *Estilo de mesclagem*
- **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 de tradução*
- **enable_suggestions** (*string*) – *Habilitar sugestões*
- **suggestion_voting** (*string*) – *Votação de sugestões*
- **suggestion_autoaccept** (*string*) – *Aceitar sugestões automaticamente*
- **push_on_commit** (*string*) – *Push ao fazer commit*
- **commit_pending_age** (*string*) – *Idade das alterações para fazer commit*
- **auto_lock_error** (*string*) – *Bloquear em erro*
- **language_regex** (*string*) – *Filtro de idioma*
- **variant_regex** (*string*) – *Expressão regular de variantes*
- **repository_url** (*string*) – URL para status 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 status 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",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
```

(continua na próxima página)

(continuação da página anterior)

```

        "name": "English",
        "url": "http://example.com/api/languages/en/",
        "web_url": "http://example.com/languages/en/"
    },
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
},
"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 amigável do projeto
- **component** (*string*) – URL amigável 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 amigável do projeto
- **component** (*string*) – URL amigável do componente

Objeto JSON da solicitação

- **branch** (*string*) – Ramo do repositório VCS
- **file_format** (*string*) – formato de arquivo das traduções
- **filemask** (*string*) – máscara dos arquivos de tradução no repositório
- **name** (*string*) – nome do componente
- **slug** (*string*) – URL amigável do componente
- **repo** (*string*) – URL do repositório VCS
- **template** (*string*) – arquivo base para traduções monolíngues
- **new_base** (*string*) – arquivo base para adicionar novas traduções

- **vcs** (*string*) – sistema de controle de versão

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

Exclui um componente.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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/` aceitando os mesmos parâmetros.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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 tela do componente.

Parâmetros

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

Objeto JSON de resposta

- **results** (*array*) – vetor de capturas de tela do componente; veja `GET /api/screenshots/ (int:id) /`

GET `/api/components/ (string: project) /`
string: `component/lock/` Retorna o status do bloqueio do componente.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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 status de bloqueio do componente.

A resposta é a mesma que `GET /api/components/ (string:project) / (string:component) /lock/`.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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 status do repositório VCS.

A resposta é a mesma para **GET** /api/projects/(string:project)/repository/.

Parâmetros

- **project** (string) – URL amigável do projeto
- **component** (string) – URL amigável 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) – Status do repositório VCS como relatado pelo VCS
- **merge_failure** – Texto descrevendo uma falha de mesclagem ou nulo se houver nenhuma

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

string: component/repository/ Executa a operação fornecida em um repositório VCS.

Veja **POST** /api/projects/(string:project)/repository/ para a documentação.

Parâmetros

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

- **component** (*string*) – URL amigável 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/* Faz o download do arquivo base para traduções monolíngues.

Parâmetros

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

GET /api/components/(string: *project*) /
string: *component/new_template/* Faz o download do arquivo de modelo para novas traduções.

Parâmetros

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

GET /api/components/(string: *project*) /
string: *component/translations/* Retorna uma lista de objetos de tradução no componente fornecido.

Parâmetros

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

- **component** (*string*) – URL amigável 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 amigável do projeto
- **component** (*string*) – URL amigável 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 objeto 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
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,
```

(continua na próxima página)

```

"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 amigável do projeto
- **component** (*string*) – URL amigável 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 amigável do projeto
- **component** (*string*) – URL amigável 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 amigável do projeto
- **component** (*string*) – URL amigável 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 amigável do projeto
- **component** (*string*) – URL amigável do componente
- **project_slug** (*string*) – Slug do projeto para remover

1.12.9 Traduções

GET `/api/translations/`

Retorna uma lista de traduções.

Ver 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 / Retorna informações sobre uma tradução.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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*) – número de textos contendo verificações com falha
- **failing_checks_percent** (*float*) – porcentagem de textos contendo verificações com falha
- **failing_checks_words** (*int*) – número de palavras contendo verificações com falha
- **filename** (*string*) – Nome de arquivo da tradução
- **fuzzy** (*int*) – número de textos aproximados (marcados para edição)
- **fuzzy_percent** (*float*) – porcentagem de textos aproximados (marcados para edição)
- **fuzzy_words** (*int*) – número de palavras em textos aproximados (marcados para edição)
- **have_comment** (*int*) – número de textos com comentário
- **have_suggestion** (*int*) – número de textos 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 arquivo
- **share_url** (*string*) – URL para compartilhar levando à página de engajamento
- **total** (*int*) – número total de textos
- **total_words** (*int*) – número total de palavras
- **translate_url** (*string*) – URL para tradução
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **translated_words** (*int*) – número de palavras traduzidas
- **repository_url** (*string*) – URL para status do repositório; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
- **file_url** (*string*) – URL para objeto arquivo; 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 textos; 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",
      "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",
```

(continua na próxima página)

(continuação da página anterior)

```

    "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.

Exclui uma tradução.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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 aceitando os mesmos parâmetros.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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 amigável do projeto
- **component** (*string*) – URL amigável 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 amigável do projeto
- **component** (*string*) – URL amigável 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

Ver também:

[Gerenciar textos](#), [adding-new-strings](#)

POST /api/translations/ (string: project) /
string: component/string: language/autotranslate/ Aciona a tradução automática.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON da solicitação

- **mode** (*string*) – Modo de tradução automática
- **filter_type** (*string*) – Tipo de filtro da tradução automática
- **auto_source** (*string*) – Fonte da tradução automática
- **component** (*string*) – Desativa contribuição para a memória de tradução compartilhada do projeto para obter acesso a componentes adicionais.
- **engines** (*string*) – Mecanismos de tradução de máquina
- **threshold** (*string*) – Limite de pontuação

GET /api/translations/ (string: project) /
string: component/string: language/file/ Baixa o arquivo de tradução atual como está armazenado no VCS (sem o parâmetro `format`) ou convertido para outro formato (veja [Baixando traduções](#)).

Nota: Este ponto de extremidade da API usa uma lógica diferente para a saída do resto da API, pois opera no arquivo inteiro em vez de nos dados. O conjunto de parâmetros aceitos no `format` difere e sem tal parâmetro você obtém o arquivo de tradução armazenado no VCS.

Parâmetros de consulta

- **format** – Formato de arquivo para usar; se não especificado nenhuma conversão de formato acontece; formatos de arquivo suportados: po, mo, xliff, xliff11, tbx, csv, xlsx, json, aresource, strings

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável do componente
- **language** (*string*) – Código de idioma de tradução

POST /api/translations/(**string**: *project*) /
string: *component*/**string**: *language*/**file**/ Envia novo arquivo com traduções.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável 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** – Arquivo 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 textos aproximados (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 status do repositório VCS.

A resposta é a mesma que para **GET** /api/components/(**string**:*project*) /(**string**:*component*)/**repository**/.

Parâmetros

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

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

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

GET /api/translations/ (**string:** *project*) /
string: *component* / **string:** *language/statistics* / Retorna as estatísticas de tradução detalhadas.

Novo na versão 2.7.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON de resposta

- **code** (*string*) – código de idioma
- **failing** (*int*) – número de verificações com falha
- **failing_percent** (*float*) – porcentagem de verificações com falha
- **fuzzy** (*int*) – número de textos aproximados (marcados para edição)
- **fuzzy_percent** (*float*) – porcentagem de textos aproximados (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*) – número total de textos
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **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 um texto traduzido 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.

Ver 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 textos plurais.

Retorna informações 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*) – texto fonte
- **previous_source** (*string*) – texto fonte anterior usado para correspondência aproximada
- **target** (*array*) – texto alvo
- **id_hash** (*string*) – identificador único da unidade
- **content_hash** (*string*) – identificador único do texto 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 marcada com “fuzzy” ou para revisão
- **translated** (*boolean*) – se a unidade está traduzida
- **approved** (*boolean*) – se a tradução está aprovada
- **position** (*int*) – posição da unidade no arquivo de tradução
- **has_suggestion** (*boolean*) – se a unidade tem sugestões
- **has_comment** (*boolean*) – se a unidade tem comentários
- **has_failing_check** (*boolean*) – se a unidade tem verificações com falha
- **num_words** (*int*) – número de palavras fonte
- **priority** (*int*) – prioridade de tradução. 100 é o padrão
- **id** (*int*) – identificador da unidade
- **explanation** (*string*) – Expansão de texto, disponível em unidades fonte, veja [Informações adicionais sobre textos fonte](#)
- **extra_flags** (*string*) – Marcadores de textos adicionais, disponíveis nas unidades fonte, veja [Personalizando comportamento usando marcadores](#)
- **web_url** (*string*) – URL onde a unidade pode ser editada
- **souce_unit** (*string*) – Link da unidade fonte; veja `GET /api/units/(int:id) /`

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

Novo na versão 4.3.

Executa 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 habilitado, veja [Revisores dedicados](#))
- **target** (*array*) – texto alvo
- **explanation** (*string*) – Expansão de texto, disponível em unidades fonte, veja [Informações adicionais sobre textos fonte](#)
- **extra_flags** (*string*) – Marcadores de textos adicionais, disponíveis nas unidades fonte, veja [Personalizando comportamento usando marcadores](#)

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

Novo na versão 4.3.

Executa atualização completa 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 habilitado, veja [Revisores dedicados](#))
- **target** (*array*) – texto alvo
- **explanation** (*string*) – Expansão de texto, disponível em unidades fonte, veja [Informações adicionais sobre textos fonte](#)
- **extra_flags** (*string*) – Marcadores de textos adicionais, disponíveis nas unidades fonte, veja [Personalizando comportamento usando marcadores](#)

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

Novo na versão 4.3.

Exclui uma 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.

Ver também:Os atributos do objeto de alteração estão documentados em `GET /api/changes/ (int:id) /`.**Parâmetros de consulta**

- **user** (*string*) – Nome de usuário do usuário para filtrar
- **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) /`

Retorna informações sobre alteração de 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 usuário 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 de texto da ação
- **target** (*string*) – texto ou detalhe alterado de evento
- **id** (*int*) – identificador da alteração

1.12.12 Capturas de tela

Novo na versão 2.14.

GET `/api/screenshots/`

Retorna uma lista de informações de texto de capturas de tela.

Ver também:

Os atributos do objeto de captura de tela estão documentados em `GET /api/screenshots/(int:id) /`.

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

Retorna informações sobre dados de captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela

Objeto JSON de resposta

- **name** (*string*) – nome de uma captura de tela
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para baixar um arquivo; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – link para associar informações sobre texto fonte; veja `GET /api/units/(int:id) /`

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

Baixa a imagem de captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela

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

Substitui a imagem de captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela

Parâmetros de formulário

- **file image** – Arquivo 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 texto fonte com uma captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela

Parâmetros de formulário

- **string unit_id** – ID da unidade

Objeto JSON de resposta

- **name** (*string*) – nome de uma captura de tela
- **translation** (*string*) – URL de um objeto de tradução relacionado
- **file_url** (*string*) – URL para baixar um arquivo; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – link para associar informações sobre texto fonte; veja `GET /api/units/(int:id)/`

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

int: *unit_id* Remove a associação de texto fonte com captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela
- **unit_id** – ID de unidade do texto fonte

POST /api/screenshots/

Cria uma nova captura de tela.

Parâmetros de formulário

- **file image** – Arquivo enviado
- **string name** – Nome da captura de tela
- **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 uma captura de tela
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para baixar um arquivo; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – link para associar informações sobre texto fonte; veja `GET /api/units/(int:id)/`

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

Edita informações parciais sobre captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela

Objeto JSON de resposta

- **name** (*string*) – nome de uma captura de tela
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para baixar um arquivo; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – link para associar informações sobre texto fonte; veja `GET /api/units/(int:id)/`

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

Edita informações completas sobre captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela

Objeto JSON de resposta

- **name** (*string*) – nome de uma captura de tela
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para baixar um arquivo; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – link para associar informações sobre texto fonte; veja `GET /api/units/(int:id)/`

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

Exclui captura de tela.

Parâmetros

- **id** (*int*) – ID da captura de tela

1.12.13 Extensões

Novo na versão 4.4.1.

GET `/api/addons/`

Retorna uma lista de extensões.

Ver 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

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) /`
Exclui extensão.

Parâmetros

- **id** (*int*) – ID da extensão

1.12.14 Listas dos componentes

Novo na versão 4.0.

GET `/api/component-lists/`
Retorna uma lista de listas de componentes.

Ver 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 em um painel

- **components** (*array*) – link 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 em um 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 em um painel

DELETE `/api/component-lists/(str: slug) /`
Exclui 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 textos comuns. Use a respectiva 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 porcentagem
- **result** (*object*) – Resultado da tarefa ou detalhes do progresso
- **log** (*string*) – Registros da tarefa

1.12.17 Ganchos de notificação

Os ganchos de notificação permitem que aplicativos externos notifiquem o Weblate que o repositório VCS foi atualizado.

Você 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 (baixando do VCS e procurando 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 em um projeto (baixando do VCS e procurando 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: habilite o gancho do serviço Weblate nas configurações do repositório e defina a URL para a URL de sua instalação do Weblate.

Ver também:

Recebendo automaticamente alterações do GitHub 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 habilitar ganchos para todo o Weblate

POST /hooks/gitlab/

Gancho especial para lidar com notificações do GitLab e atualizar automaticamente os componentes correspondentes.

Ver também:

Recebendo automaticamente alterações do GitLab 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 habilitar ganchos para todo o Weblate

POST /hooks/bitbucket/

Gancho especial para lidar com notificações do Bitbucket e atualizar automaticamente os componentes correspondentes.

Ver também:

Recebendo automaticamente alterações do Bitbucket 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 habilitar 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.

Ver também:

Recebendo automaticamente alterações do Pagure 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 habilitar 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.

Ver também:

Recebendo automaticamente alterações do Azure Repos 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 habilitar 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.

Ver também:

Recebendo automaticamente alterações do Gitea Repos Para instruções sobre como configurar integração com Gitea

<https://docs.gitea.io/pt-br/webhooks/> Informações genéricas sobre Webhooks do Gitea

ENABLE_HOOKS Para habilitar 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.

Ver também:

Recebendo automaticamente alterações de Gitee Repos Para instruções sobre como configurar integração com Gitee

<https://gitee.com/help/categories/40> Informações genéricas sobre Webhooks do Gitee

ENABLE_HOOKS Para habilitar ganchos para todo o Weblate

1.12.18 Exportações

O Weblate fornece várias exportações para permitir que você 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,
```

(continua na próxima página)

(continuação da página anterior)

```

    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "last_author": "Michal Čihař",
    "last_change": "2012-03-28T15:07:38+00:00",
    "name": "Czech",
    "total": 436,
    "total_words": 15271,
    "translated": 436,
    "translated_percent": 100.0,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/cs/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/cs/"
  },
  {
    "code": "nl",
    "failing": 21,
    "failing_percent": 4.8,
    "fuzzy": 11,
    "fuzzy_percent": 2.5,
    "last_author": null,
    "last_change": null,
    "name": "Dutch",
    "total": 436,
    "total_words": 15271,
    "translated": 319,
    "translated_percent": 73.2,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/nl/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/nl/"
  },
  {
    "code": "el",
    "failing": 11,
    "failing_percent": 2.5,
    "fuzzy": 21,
    "fuzzy_percent": 4.8,
    "last_author": null,
    "last_change": null,
    "name": "Greek",
    "total": 436,
    "total_words": 15271,
    "translated": 312,
    "translated_percent": 71.6,
    "translated_words": 3201,
    "url": "http://hosted.weblate.org/engage/weblate/el/",
    "url_translate": "http://hosted.weblate.org/projects/weblate/main/el/"
  }
]

```

1.12.19 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.

Ver também:

[RSS na Wikipédia](#)

1.13 Weblate Client

Novo na versão 2.7: Há suporte total do utilitário `wlc` desde o Weblate 2.7. Se você estiver usando uma versão mais antiga, algumas incompatibilidades com a API podem ocorrer.

1.13.1 Instalação

O Weblate Client é enviado separadamente e inclui o módulo Python. Para usar os comandos abaixo, você precisa instalar `wlc`:

```
pip3 install wlc
```

1.13.2 Uso com 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>

Instalação:

```
docker pull weblate/wlc
```

O contêiner 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 contêiner 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
```

Você pode querer passar seu [Arquivos de configuração](#) para o contêiner Docker, a abordagem mais fácil é adicionar 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` (consulte [Arquivos de configuração](#) para outros locais). Crie-o para corresponder ao seu ambiente:

```
[weblate]
url = https://hosted.weblate.org/api/

[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Em seguida, você pode invocar comandos no servidor padrão:

```
wlc ls
wlc commit sandbox/hello-world
```

Ver também:

[Arquivos de configuração](#)

1.13.4 Sinopse

```
wlc [arguments] <command> [options]
```

Os comandos indicam, na verdade, qual operação deve ser realizada.

1.13.5 Descrição

Weblate Client é uma biblioteca Python e utilitário de linha de comando para gerenciar o Weblate remotamente usando 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 instância do Weblate usar. Estes devem ser inseridos 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 arquivo de configuração, consulte [Arquivos de configuração](#). A URL deve terminar com `/api/`, por exemplo, `https://hosted.weblate.org/api/`.

--key KEY
Especifica a chave do usuário de API a ser usada. Substitui qualquer valor encontrado no arquivo de configuração, consulte [Arquivos de configuração](#). Você pode encontrar sua chave em seu perfil no Weblate.

--config PATH
Substitui o caminho do arquivo de configuração, consulte [Arquivos de configuração](#).

--config-section SECTION
Substitui a seção de arquivos de configuração em uso, consulte [Arquivos de configuração](#).

Comandos

Os seguintes comandos 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 commit das alterações feitas em um objeto Weblate (tradução, componente ou projeto).

pull

Faz pull das alterações remotas do repositório no objeto Weblate (tradução, componente ou projeto).

push

Faz push das alterações do objeto Weblate para o 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 quaisquer alterações não rastreadas em um objeto Weblate para corresponder ao repositório remoto (tradução, componente ou projeto).

repo

Exibe o status do repositório para um determinado objeto 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 status 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.

Baixa um arquivo de tradução.

--convert

Converte o formato do arquivo, se nenhuma conversão não especificada acontecer no servidor e o arquivo for baixado como está no repositório.

--output

Especifica o arquivo para salvar a saída, se não for especificado, ele é impresso na stdout (saída padrão).

upload

Novo na versão 0.9: Suportado desde o wlc 0.9.

Baixa um arquivo de tradução.

--overwrite

Substitui as traduções existentes ao enviar.

--input

Arquivo a partir do qual o conteúdo é lido, se não especificado é lido a partir de stdin (entrada padrão).

Dica: Você pode obter informações mais detalhadas sobre como invocar comandos individuais passando `--help`, por exemplo: `wlc ls --help`.

1.13.6 Arquivos de configuração

.weblate, **.weblate.ini**, **weblate.ini** Alterado na versão 1.6: Os arquivos com extensão *.ini* também são aceitos.

Por arquivo de configuração de projeto

C:\Users\NAME\AppData\weblate.ini Novo na versão 1.6.

Arquivo de configuração do usuário no Windows.

~/ .config/weblate Arquivo de configuração do usuário

/etc/xdg/weblate Arquivo de configuração para todo o sistema

O programa segue a especificação XDG, para que você possa ajustar a colocação de arquivos de configuração por variáveis de ambiente `XDG_CONFIG_HOME` ou `XDG_CONFIG_DIRS`. No Windows, o diretório `APPDATA` é o local preferido para o arquivo de configuração.

As configurações seguintes podem ser configuradas na seção `[weblate]` (você pode personalizar isso por `--config-section`):

key

Chave de API para acessar o Weblate.

url

URL de API do servidor, com o padrão sendo `http://127.0.0.1:8000/api/`.

translation

Caminho para a tradução padrão - componente ou projeto.

O arquivo de configuração é um arquivo 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 seção `[keys]`:

```
[keys]
https://hosted.weblate.org/api/ = APIKEY
```

Isso permite que você armazene chaves em suas configurações pessoais, enquanto usa a configuração do `.weblate` no repositório VCS para que o `wlc` saiba com qual servidor ele deve falar.

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/
```

Você 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 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, você precisa instalar o *Weblate Client*: (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 usuário
- **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 arquivo de configuração seguindo a especificação XDG.

load (*path*=None)

Parâmetros `path` (*str*) – Caminho a partir do qual carregar a configuração.

Carrega a configuração de um arquivo. Se nenhuma for especificada, carrega do arquivo de configuração `wlc` (`~/ .config/wlc`) colocado em 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 arquivo 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 Instalando o Weblate

Instalando usando Docker

Com a implantação do Weblate dockerizada, você pode colocar 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 em um 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 arquivos, banco de dados e Weblate).

Muitos usuários 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 seu tamanho mínimo fazendo clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções gerenciadas nele.

Instalação

Os exemplos a seguir presumem que você 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 arquivo `docker-compose.override.yml` com 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 usuário `admin` é criado com uma senha aleatória mostrada na primeira inicialização.

O exemplo fornecido faz o Weblate escutar na porta 80. Edite o mapeamento da porta no arquivo `docker-compose.override.yml` para alterar isso.

3. Inicie os contêineres do Weblate:

```
docker-compose up
```

Aproveite a implantação do Weblate, ele está acessível na porta 80 do contêiner `weblate`.

Alterado na versão 2.15-2: A configuração foi alterada recentemente, antes havia um contêiner de servidor web separado, desde 2.15-2 o servidor web está embutido no contêiner do Weblate.

Alterado na versão 3.7.1-6: Em julho de 2019 (começando com a tag 3.7.1-6), os contêineres não estão sendo executados como um usuário `root`. Isso mudou a porta exposta de 80 para 8080.

Ver também:

Invocando comandos de gerenciamento

Contêiner Docker com suporte a HTTPS

Por favor, veja [Instalação](#) para instruções genéricas de implantação, esta seção apenas menciona diferenças em comparação a ela.

Usando seus próprios certificados SSL

Novo na versão 3.8-3.

No caso de você ter seu próprio certificado SSL que deseja usar, basta colocar os arquivos no volume de dados Weblate (veja [Volumes de contêiner Docker](#)):

- `ssl/fullchain.pem` contendo o certificado, incluindo quaisquer certificados CA necessários
- `ssl/privkey.pem` contendo a chave privada

Ambos os arquivos devem pertencer ao mesmo usuário que inicia o contêiner do docker e ter a máscara de arquivo definida como 600 (legível e gravável apenas pelo usuário dono).

Além disso, o contêiner Weblate agora aceitará conexões SSL na porta 4443. Você ainda vai querer incluir o encaminhamento de porta para HTTPS na substituição de composição do docker:

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
      - 443:4443
```

Se você 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 contêiner do docker, você 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 de seu ambiente.

Certificados SSL automáticos usando Let's Encrypt

Caso você queira usar certificados SSL [Let's Encrypt](#) gerados automaticamente na instalação pública, você precisa adicionar um proxy HTTPS reverso em um contêiner Docker adicional, [https-portal](#) será usado para isso. Isso é usado no arquivo `docker-compose-https.yml`. Em seguida, crie um arquivo `docker-compose-https.override.yml` com 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**, você precisa passar os dois arquivos para 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 contêiner Docker

Normalmente, é uma boa ideia atualizar apenas o contêiner Weblate e manter o contêiner PostgreSQL na versão que você possui, já que atualizar o PostgreSQL é muito doloroso e na maioria dos casos não traz muitos benefícios.

Você pode fazer isso mantendo 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 você 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 atualizando para as versões mais recentes.

Você também pode querer 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 contêiner, você pode entrar como usuário *admin* com a senha fornecida em `WEBLATE_ADMIN_PASSWORD`, ou uma senha aleatória gerada na primeira inicialização se não tiver sido definida.

Para redefinir a senha do *admin*, reinicie o contêiner com `WEBLATE_ADMIN_PASSWORD` definido com a nova senha.

Ver também:

`WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

Number of processes and memory consumption

The number of worker processes for both uWSGI and Celery is determined automatically based on number of CPUs. This works well for most cloud virtual machines as these typically have few CPUs and good amount of memory.

Caso você tenha muitos núcleos de CPU e tenha problemas de memória insuficiente, tente reduzir o número de workers:

```
environment:
  WEBLATE_WORKERS: 2
```

You can also fine-tune individual worker categories:

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

Ver também:

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

Scaling horizontally

Novo na versão 4.6.

Aviso: This feature is a technology preview.

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

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

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

Variáveis de ambiente do Docker

Muitas das *Configurações* do Weblate podem ser definidas no contêiner Docker usando variáveis de ambiente:

Configurações genéricas

WEBLATE_DEBUG

Configura o modo de depuração do Django usando *DEBUG*.

Exemplo:

```
environment:
  WEBLATE_DEBUG: 1
```

Ver também:

Desabilitar 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.

Ver 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 usuário *admin* (veja *WEBLATE_ADMIN_PASSWORD* para mais informações).

Exemplo:

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

Ver também:

Autenticação como administrador, Configurar corretamente administradores, ADMINS

WEBLATE_ADMIN_PASSWORD

Define a senha para o usuário *admin*.

- Se não for definido e o usuário *admin* não existir, ele será criado com uma senha aleatória mostrada na primeira inicialização do contêiner.
- Se não for definido e o usuário *admin* existir, nenhuma ação será executada.
- Se definido, o usuário *admin* é ajustado em cada inicialização do contêiner para corresponder a *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_NAME* e *WEBLATE_ADMIN_EMAIL*.

Aviso: Pode ser um risco de segurança armazenar a senha no arquivo de configuração. Considere usar essa variável apenas para configuração inicial (ou deixe o Weblate gerar uma senha aleatória na inicialização) ou para recuperação de senha.

Ver também:

Autenticação como administrador, `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

WEBLATE_SERVER_EMAIL**WEBLATE_DEFAULT_FROM_EMAIL**

Configura o endereço para e-mails de saída.

Ver também:

Configurar envio de e-mail

WEBLATE_CONTACT_FORM

Configures contact form behavior, see `CONTACT_FORM`.

WEBLATE_ALLOWED_HOSTS

Configura os nomes de host HTTP permitidos usando `ALLOWED_HOSTS`.

O padrão é ```*``` que permite todos os nomes de host.

Exemplo:

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

Ver também:

`ALLOWED_HOSTS`, Configuração de hosts permitidos, Definir domínio correto do site

WEBLATE_REGISTRATION_OPEN

Configura se os registros são abertos alternando `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 contêiner do Docker, use a variável de ambiente ```TZ```.

Exemplo:

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

WEBLATE_ENABLE_HTTPS

Faz com que o Weblate presuma que é operado por trás de um proxy HTTPS reverso, faz com que o Weblate use HTTPS em e-mail e links 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 contêiner Weblate aceite conexões HTTPS, você precisa configurar isso também, consulte [Contêiner Docker com suporte a HTTPS](#) para exemplos.

Exemplo:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Ver 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 contêiner Weblate.

Habilita [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 sendo executado por trás de um proxy reverso fazendo a terminação SSL que não passa cabeçalhos HTTPS padrão.

Exemplo:

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

Ver também:

[SECURE_PROXY_SSL_HEADER](#)

WEBLATE_REQUIRE_LOGIN

Habilita [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 usando *LOGIN_REQUIRED_URLS_EXCEPTIONS*.

Você pode substituir configurações inteiras ou modificar o valor padrão usando as variáveis ADD e REMOVE.

WEBLATE_GOOGLE_ANALYTICS_ID

Configura o ID para o Google Analytics alterando *GOOGLE_ANALYTICS_ID*.

WEBLATE_GITHUB_USERNAME

Configura o nome de usuário do GitHub para pull requests do GitHub alterando *GITHUB_USERNAME*.

Ver 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 alterando *GITHUB_TOKEN*.

Ver também:

GitHub

WEBLATE_GITLAB_USERNAME

Configura o nome de usuário do GitLab para merge requests do GitLab alterando *GITLAB_USERNAME*

Ver também:

GitLab

WEBLATE_GITLAB_TOKEN

Configura o token de acesso pessoal do GitLab para merge requests do GitLab via API alterando *GITLAB_TOKEN*

Ver também:

GitLab

WEBLATE_PAGURE_USERNAME

Configura o nome de usuário do Pagure para merge requests do Pagure alterando *PAGURE_USERNAME*

Ver também:

Pagure

WEBLATE_PAGURE_TOKEN

Configura o token de acesso pessoal do Pagure para merge requests do Pagure via API alterando *PAGURE_TOKEN*

Ver também:

Pagure

WEBLATE_SIMPLIFY_LANGUAGES

Configura a política de simplificação de idioma, veja *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configura o padrão *Controle 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 de 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*.

Ver também:

Signing Git commits with GnuPG

WEBLATE_URL_PREFIX

Configura o prefixo da URL onde o Weblate está sendo executado, veja *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Configura verificações que você 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.

Ver 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: You can set configuration for any rate limiter scopes. To do that add `WEBLATE_` prefix to any of setting described in *Limitação de taxa*.

Ver também:

Limitação de taxa, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

WEBLATE_ENABLE_AVATARS

Novo na versão 4.6.1.

Configures `ENABLE_AVATARS`.

Configurações de tradução automática

WEBLATE_MT_APERTIUM_APY

Habilita 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

Habilita tradução de máquina do *DeepL* e define `MT_DEEPL_KEY`

WEBLATE_MT_DEEPL_API_VERSION

Configura a versão da API do *DeepL* para usar, veja `MT_DEEPL_API_VERSION`.

WEBLATE_MT_GOOGLE_KEY

Habilita *Google Translate* e define `MT_GOOGLE_KEY`

WEBLATE_MT_MICROSOFT_COGNITIVE_KEY

Habilita *Microsoft Cognitive Services Translator* 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

Define `MT_MICROSOFT_REGION`

WEBLATE_MT_MICROSOFT_BASE_URL

Define `MT_MICROSOFT_BASE_URL`

WEBLATE_MT_MODERNMT_KEY

Habilita *ModernMT* e define `MT_MODERNMT_KEY`.

WEBLATE_MT_MYMEMORY_ENABLED

Habilita 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

Habilita tradução de máquina do *Glosbe*.

```
environment:
  WEBLATE_MT_GLOSBE_ENABLED: 1
```

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED
Habilita tradução de máquina do *Microsoft Terminology Service*.

```
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.
  # 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)
```

Ver também:

Autenticação por LDAP

GitHub

WEBLATE_SOCIAL_AUTH_GITHUB_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_SECRET

Habilita *Autenticação por GitHub*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Habilita *Autenticação por Bitbucket*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Habilita *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`

Habilita *OAuth 2 do Google*.

GitLab

`WEBLATE_SOCIAL_AUTH_GITLAB_KEY`
`WEBLATE_SOCIAL_AUTH_GITLAB_SECRET`
`WEBLATE_SOCIAL_AUTH_GITLAB_API_URL`

Habilita *OAuth 2 do GitLab*.

Active Directory do Azure

`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY`
`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET`

Habilita 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`

Habilita 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`

Habilita autenticação com Keycloak, veja a [documentação](#).

Fornecedores Linux

Você pode habilitar a autenticação usando serviços de autenticação de fornecedores Linux, definindo 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

Habilita a autenticação Slack, veja [Slack](#).

SAML

Chaves SAML autoassinadas são geradas automaticamente na primeira inicialização do contêiner. Caso você queira usar chaves próprias, coloque 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

Desabilita 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 contêineres Weblate e PostgreSQL.

Ver também:

Configuração de banco de dados para o Weblate

POSTGRES_PASSWORD

Senha do PostgreSQL.

POSTGRES_USER

Nome de usuário 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 [Configurando Weblate para usar PostgreSQL](#).

Configurações de backup de banco de dados

Ver também:

Dados despejados para os backups

WEBLATE_DATABASE_BACKUP

Configura o despejo diário do banco de dados usando [DATABASE_BACKUP](#). O padrão é `plain`.

Configuração do servidor de cache

O uso do Redis é altamente recomendado pelo Weblate e você deve fornecer uma instância do Redis ao executar o Weblate no Docker.

Ver também:

Habilitar 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 senha do servidor Redis, não usada por padrão.

REDIS_TLS

Habilita 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, você 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
```

(continua na próxima página)

(continuação da página anterior)

```

WEBLATE_EMAIL_HOST_PASSWORD: pass
WEBLATE_EMAIL_USE_TLS: 0
WEBLATE_EMAIL_USE_SSL: 1

```

Ver também:*Configuração de e-mail de saída***WEBLATE_EMAIL_HOST**

Nome de host ou endereço IP do servidor de correio.

Ver 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.

Ver também:*EMAIL_PORT***WEBLATE_EMAIL_HOST_USER**

Usuário da autenticação por e-mail.

Ver também:*EMAIL_HOST_USER***WEBLATE_EMAIL_HOST_PASSWORD**

Senha da autenticação por e-mail.

Ver também:*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 você estiver tendo problemas, consulte a configuração TLS explícita *WEBLATE_EMAIL_USE_TLS*.

Ver 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 você estiver tendo conexões travadas, consulte a configuração TLS implícita *WEBLATE_EMAIL_USE_SSL*.

Ver 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.

Ver 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 habilitar o suporte para Rollbar, defina o seguinte:

ROLLBAR_KEY

Seu token de acesso ao servidor de postagem Rollbar.

ROLLBAR_ENVIRONMENT

Seu ambiente Rollbar, o padrão é *production*.

Para habilitar o suporte para Sentry, defina o seguinte:

SENTRY_DSN

Seu DSN no Sentry.

SENTRY_ENVIRONMENT

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 contêiner. 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: Você é responsável por configurar o serviço dos arquivos gerados pelo Weblate, ele só armazena os arquivos no local configurado.

Ver também:

weblate-cdn, *LOCALIZE_CDN_URL*, *LOCALIZE_CDN_PATH*

Alterando aplicativos, verificações, complementos ou correções automáticas habilitados

Novo na versão 3.8-5.

A configuração embutida de verificações, complementos ou correções automática habilitados 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
```

Ver também:

CHECK_LIST, AUTOFIX_LIST, WEBLATE_ADDONS, INSTALLED_APPS

Configurações do contêiner

WEBLATE_WORKERS

Novo na versão 4.6.1.

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

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

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

Essas variáveis permitem que você ajuste as opções do worker do Celery. Pode ser útil ajustar a simultaneidade (`--concurrency 16`) ou usar diferentes implementações de pool (`--pool=gevent`).

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

Exemplo:

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

Ver também:

[Opções de worker do Celery](#), [Tarefas de fundo usando Celery](#)

UWSGI_WORKERS

Configura quantos workers uWSGI devem ser executados.

It defaults to [WEBLATE_WORKERS](#).

Exemplo:

```
environment:  
  UWSGI_WORKERS: 32
```

WEBLATE_SERVICE

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

Following services are defined:

celery-beat Celery task scheduler, only one instance should be running. This container is also responsible for the database structure migrations and it should be started prior others.

celery-backup Celery worker for backups, only one instance should be running.

celery-celery Generic Celery worker.

celery-memory Translation memory Celery worker.

celery-notify Notifications Celery worker.

celery-translate Automatic translation Celery worker.

web Web server.

Volumes de contêiner Docker

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

O volume de dados é usado para armazenar dados persistentes do Weblate, como repositórios clonados ou para personalizar a instalação do Weblate.

O posicionamento do volume Docker no sistema hospedeiro depende da configuração do Docker, mas geralmente é armazenado em `/var/lib/docker/volumes/weblate-docker_weblate-data/_data/`. No contêiner, ele é montado como `/app/data`.

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

Ver também:

[Documentação sobre volumes do Docker](#)

Personalização adicional da configuração

Você pode personalizar ainda mais a instalação do Weblate no volume de dados, veja [Volumes de contêiner Docker](#).

Arquivos de configuração personalizados

Você também pode sobrescrever a configuração em `/app/data/settings-override.py` (veja *Volumes de contêiner Docker*). Isso é executado no final das configurações embutidas, depois que todas as configurações de ambiente são carregadas e você pode ajustá-las ou substituí-las.

Substituindo o logotipo e outros arquivos estáticos

Novo na versão 3.8-5.

Os arquivos estáticos que vêm com Weblate podem ser sobrescritos colocando em `/app/data/python/customize/static` (veja *Volumes de contêiner Docker*). Por exemplo, criar `/app/data/python/customize/static/favicon.ico` substituirá o `favicon`.

Dica: Os arquivos são copiados para o local correspondente na inicialização do contêiner, portanto, é necessário reiniciar o Weblate após alterar o conteúdo do volume.

Como alternativa, você também pode incluir o próprio módulo (veja `../customize`) e adicioná-lo como um volume separado ao contêiner 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 seus próprios módulos Python

Novo na versão 3.8-5.

Você pode colocar os próprios módulos Python em `/app/data/python/` (veja *Volumes de contêiner Docker*) e eles podem ser carregados pelo Weblate, provavelmente usando `docker-custom -config`.

Ver também:

Personalizando o Weblate

Selecione sua máquina – provedores locais ou em nuvem

Com Docker Machine, você pode criar sua implantação Weblate em sua máquina local ou em qualquer grande número 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 em um 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 arquivos, banco de dados e Weblate).

Muitos usuários 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 seu tamanho mínimo fazendo clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções gerenciadas 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, dependendo dos recursos que você pretende usar (veja *Dependências opcionais*):

```
apt install tesseract-ocr libtesseract-dev liblibleptonica-dev
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executando servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo usando Celery*. Dependendo do tamanho da sua instalação, você pode querer 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

# SMTP server
apt install exim4
```

Módulos do Python

Dica: Estamos usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você 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 dependendo dos recursos que você 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, você terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você pode ficar com o exemplo enviado para fins de teste, mas você vai querer mudanças para configuração de produção, veja [Ajustando a configuração](#).

3. Crie o banco de dados e sua estrutura 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 usuário do administrador e copie a senha de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete arquivos estáticos para o servidor web (veja [Executando servidor](#) e [Servindo arquivos estáticos](#)):

```
weblate collectstatic
```

6. Compactar arquivos JavaScript e CSS (opcional, veja [Comprimindo 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 usando Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executando servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, seu servidor Weblate agora está funcionando e você pode começar a usá-lo.

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou usando **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 você precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta para o repositório VCS e seleciona quais arquivos traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: Nome do componente, endereço do repositório VCS e máscara para localizar arquivos traduzíveis. Weblate suporta uma ampla gama de formatos, incluindo arquivos PO gettext, textos de recursos Android, propriedades de textos iOS, propriedades Java ou arquivos Qt Linguist, consulte [Formatos de arquivos suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado dependendo do tamanho do seu repositório VCS e do número de mensagens a serem traduzidas), você 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 em um 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 arquivos, banco de dados e Weblate).

Muitos usuários 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 seu tamanho mínimo fazendo clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções gerenciadas 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, dependendo dos recursos que você pretende usar (veja *Dependências opcionais*):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executando servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo usando Celery*. Dependendo do tamanho da sua instalação, você pode querer 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

# SMTP server
zypper install postfix
```

Módulos do Python

Dica: Estamos usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você 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 dependendo dos recursos que você 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, você terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você pode ficar com o exemplo enviado para fins de teste, mas você vai querer mudanças para configuração de produção, veja [Ajustando a configuração](#).

3. Crie o banco de dados e sua estrutura 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 usuário do administrador e copie a senha de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete arquivos estáticos para o servidor web (veja [Executando servidor](#) e [Servindo arquivos estáticos](#)):

```
weblate collectstatic
```

6. Compactar arquivos JavaScript e CSS (opcional, veja [Comprimindo 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 usando Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executando servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, seu servidor Weblate agora está funcionando e você pode começar a usá-lo.

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou usando **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 você precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta para o repositório VCS e seleciona quais arquivos traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: Nome do componente, endereço do repositório VCS e máscara para localizar arquivos traduzíveis. Weblate suporta uma ampla gama de formatos, incluindo arquivos PO gettext, textos de recursos Android, propriedades de textos iOS, propriedades Java ou arquivos Qt Linguist, consulte [Formatos de arquivos suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado dependendo do tamanho do seu repositório VCS e do número de mensagens a serem traduzidas), você 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 em um 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 arquivos, banco de dados e Weblate).

Muitos usuários 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 seu tamanho mínimo fazendo clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções gerenciadas 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, dependendo dos recursos que você 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 [Executando servidor](#), [Configuração de banco de dados para o Weblate](#), [Tarefas de fundo usando Celery](#). Dependendo do tamanho da sua instalação, você pode querer 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

# SMTP server
dnf install postfix
```

Módulos do Python

Dica: Estamos usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você 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 dependendo dos recursos que você 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, você terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você pode ficar com o exemplo enviado para fins de teste, mas você vai querer mudanças para configuração de produção, veja [Ajustando a configuração](#).
3. Crie o banco de dados e sua estrutura 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 usuário do administrador e copie a senha de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete arquivos estáticos para o servidor web (veja [Executando servidor](#) e [Servindo arquivos estáticos](#)):

```
weblate collectstatic
```

6. Compactar arquivos JavaScript e CSS (opcional, veja [Comprimindo 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 usando Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executando servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, seu servidor Weblate agora está funcionando e você pode começar a usá-lo.

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou usando **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 você precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta para o repositório VCS e seleciona quais arquivos traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: Nome do componente, endereço do repositório VCS e máscara para localizar arquivos traduzíveis. Weblate suporta uma ampla gama de formatos, incluindo arquivos PO gettext, textos de recursos Android, propriedades de textos iOS, propriedades Java ou arquivos Qt Linguist, consulte [Formatos de arquivos suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado dependendo do tamanho do seu repositório VCS e do número de mensagens a serem traduzidas), você 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 em um 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 arquivos, banco de dados e Weblate).

Muitos usuários 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 seu tamanho mínimo fazendo clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções gerenciadas 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, dependendo dos recursos que você pretende usar (veja *Dependências opcionais*):

```
brew install tesseract
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executando servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo usando Celery*. Dependendo do tamanho da sua instalação, você pode querer 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

# Database server: PostgreSQL
brew install postgresql
```

Módulos do Python

Dica: Estamos usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você 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 dependendo dos recursos que você 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, você terá que especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você pode ficar com o exemplo enviado para fins de teste, mas você vai querer mudanças para configuração de produção, veja [Ajustando a configuração](#).

3. Crie o banco de dados e sua estrutura 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 usuário do administrador e copie a senha de saída para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

5. Colete arquivos estáticos para o servidor web (veja [Executando servidor](#) e [Servindo arquivos estáticos](#)):

```
weblate collectstatic
```

6. Compactar arquivos JavaScript e CSS (opcional, veja [Comprimindo 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 usando Celery](#) para mais informações:

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

8. Inicie o servidor de desenvolvimento (veja [Executando servidor](#) para configuração de produção):

```
weblate runserver
```

Após a instalação

Parabéns, seu servidor Weblate agora está funcionando e você pode começar a usá-lo.

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Autentique-se com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` ou usando **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 você precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta para o repositório VCS e seleciona quais arquivos traduzir. Veja [Configuração de componente](#) para mais detalhes.

Os campos importantes aqui são: Nome do componente, endereço do repositório VCS e máscara para localizar arquivos traduzíveis. Weblate suporta uma ampla gama de formatos, incluindo arquivos PO gettext, textos de recursos Android, propriedades de textos iOS, propriedades Java ou arquivos Qt Linguist, consulte [Formatos de arquivos suportados](#) para mais detalhes.

3. Depois que o acima for concluído (pode ser um processo demorado dependendo do tamanho do seu repositório VCS e do número de mensagens a serem traduzidas), você 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 usando Git (ou baixe um tarball e descompacte-o):

```
git clone https://github.com/WeblateOrg/weblate.git weblate-src
```

Alternativamente, você pode usar arquivos lançados. Você pode baixá-los em nosso site <<https://weblate.org/>>. Esses downloads são assinados criptograficamente, consulte [Verificando 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 arquivo `settings.py` de acordo com sua preferência. Você pode ficar com o exemplo enviado para fins de teste, mas você vai querer mudanças para configuração de produção, veja [Ajustando a configuração](#).
6. Crie o banco de dados usado pelo Weblate, veja [Configuração de banco de dados para o Weblate](#).
7. Construa tabelas Django, arquivos estáticos e dados iniciais (veja [Preenchendo o banco de dados e Servindo arquivos estáticos](#)):

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

Nota: Esta etapa deve ser repetida sempre que você atualizar o repositório.

Instalando no OpenShift

Com o modelo OpenShift Weblate, você pode colocar 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.

Você pode encontrar o modelo em <https://github.com/WeblateOrg/openshift/>.

Instalação

Os exemplos a seguir presumem que você tem um ambiente OpenShift v3.x funcional, com a ferramenta cliente `oc` instalada. Confira a documentação do OpenShift para obter instruções.

Console web

Copie o conteúdo bruto de `template.yml` e importe-o para seu projeto, então use o botão **Create** no console web do OpenShift para criar seu aplicativo. 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 arquivo `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 usando o console web ou o CLI.

Parâmetros

Os parâmetros que você pode substituir estão listados na seção de parâmetros do modelo. Você pode listá-los com a CLI usando o seguinte comando e especificando o arquivo 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

Você 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 contêiner, você pode entrar como usuário *admin* com a senha fornecida em WEBLATE_ADMIN_PASSWORD, ou uma senha aleatória gerada na primeira inicialização se não tiver sido definida.

Para redefinir a senha do *admin*, reinicie o contêiner com WEBLATE_ADMIN_PASSWORD definido para a nova senha 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, você pode colocar 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.

Você 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
```

Dependendo da sua configuração e experiência, escolha um método de instalação apropriado para você:

- *Instalando usando Docker*, recomendado para configurações de produção.
- Instalação com 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 usando outros serviços para sua operação. Você precisará pelo menos seguir os 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 usando Celery*.
- Servidor SMTP para e-mail de saída, consulte *Configuração de e-mail de saída*.

Dependências Python

Weblate é escrito em [Python](#) e tem suporte a Python 3.6 ou mais novo. Você pode instalar dependências usando pip ou de seus pacotes de distribuição, 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 seguintes módulos são necessários para alguns recursos do Weblate. Você pode encontrar todos eles em `requirements-optional.txt`.

Mercurial (opcional para suporte a repositórios Mercurial) <https://www.mercurial-scm.org/>

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

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

akismet (opcional para a sugestão de proteção de spam) <https://github.com/ubernostrum/akismet>

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

Zeep (opcional para *Microsoft Terminology Service*) <https://docs.python-zeep.org/>

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

Dependências de backend de banco de dados

O Weblate tem suporte a 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 seguintes dependências devem ser instaladas no sistema:

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

Pango, Cairo e arquivos de cabeçalho relacionados e dados de introspecção gir <https://cairographics.org/>, <https://pango.gnome.org/>, veja *Pango e Cairo*

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

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

tesseract e seus dados (opcional para OCR de capturas de tela) <https://github.com/tesseract-ocr/tesseract>

licensee (opcional para detectar a licença ao criar o componente) <https://github.com/licensee/licensee>

Dependências de tempo de compilação

Para compilar alguns das *dependências Python*, você pode precisar instalar suas dependências. Isso depende de como você instalá-las, por isso consulte pacotes individuais para obter documentação. Você não precisará deles se usar *Wheels* pré-compilado durante a instalação usando *pip* ou quando você usar pacotes de distribuição.

Pango e Cairo

Alterado na versão 3.7.

O Weblate usa Pango e Cairo para renderizar widgets bitmap (ver *promotion*) e verificações de renderização (ver *Gerenciando fontes*). Para instalar corretamente as ligações Python para aqueles que você precisa instalar bibliotecas de sistemas primeiro - você precisa tanto do Cairo quanto do Pango, que por sua vez precisam de GLib. Todos esses devem ser instalados com arquivos de desenvolvimento e dados de introspecção GObject.

2.1.3 Verificando assinaturas de lançamento

Os lançamentos do Weblate é criptograficamente assinados pelo desenvolvedor que o lançou. Atualmente este é Michal Čihař. A impressão digital da chave PGP é:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

e você pode obter mais informações de identificação de <<https://keybase.io/nijel>>.

Você deve verificar se a assinatura corresponde ao arquivo que você baixou. Desta forma, você pode ter certeza de que está usando o mesmo código que foi lançado. Você também deve verificar a data da assinatura para ter certeza de que você baixou a versão mais recente.

Cada arquivo é acompanhado com arquivos `.asc`, os quais contêm a assinatura PGP para ele. Uma vez que você tenha ambos na mesma pasta, você 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 você pode ver, o GPG reclama que não conhece a chave pública. Neste ponto você deve fazer um dos seguintes passos:

- Use *wkd* para baixar 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@braiins.cz>
sub  rsa4096 2009-06-17 [E]
sub  rsa4096 2015-09-09 [S]
```

- Baixe o chaveiro do [servidor do Michal](#) e importe-o com:

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

- Baixe e importe a chave de um dos principais servidores:

```
$ 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 um pouco a situação - neste momento, você pode verificar que a assinatura da chave dada está correta, mas você 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@braiins.cz>" [ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

O problema aqui é que qualquer um poderia emitir a chave com este nome. Você precisa garantir que a chave é realmente 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 conhecer o desenvolvedor pessoalmente e trocar impressões digitais importantes, no entanto você também pode confiar na rede de confiança. Dessa forma, você pode confiar na chave transitivamente através de assinaturas de outros, que conheceram o desenvolvedor 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@braiins.cz>" [ultimate]
```

Se a assinatura for inválida (o arquivo foi alterado), você 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 Čihar <michal@cihar.com>" [ultimate]
```

2.1.4 Permissões do sistema de arquivos

O processo Weblate precisa ser capaz de ler e escrever para o diretório onde mantém os dados – `DATA_DIR`. Todos os arquivos dentro deste diretório devem ser de propriedade e graváveis pelo usuário que executa todos os processos do Weblate (geralmente WSGI e Celery, veja *Executando servidor* e *Tarefas de fundo usando Celery*).

A configuração padrão coloca-os na mesma árvore que os fontes do Weblate, no entanto, você 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.

Você também deve tomar cuidado ao executar *Comandos de gerência*, pois eles devem ser executados sob o mesmo usuário que o Weblate em si está sendo executado, caso contrário, permissões em alguns arquivos podem estar erradas.

No contêiner Docker, todos os arquivos no volume `/app/data` tem de ter como dono o usuário weblate dentro do contêiner (UID 1000).

Ver também:

Servindo arquivos estáticos

2.1.5 Configuração de banco de dados para o Weblate

Recomenda-se executar o Weblate com um servidor de banco de dados PostgreSQL.

Ver também:

Usar um poderoso mecanismo de banco de dados, Banco de Dados, 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 uma extensão de trígama que deve ser instalada separadamente em alguns casos. Procure por `postgresql-contrib` ou um pacote com nome similar.

Ver também:

PostgreSQL notas

Criando um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate em um banco de dados separado e separar a conta do usuário:

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

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

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

Dica: Se você não quiser fazer do usuário do Weblate um superusuário no PostgreSQL, você pode omitir isso. Nesse caso, você terá que executar algumas das etapas de migração manualmente como um superusuário PostgreSQL no esquema Weblate usará:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
```

Configurando Weblate para usar PostgreSQL

O trecho de `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 usuário. Em configurações mais complexas, o nome da função é diferente do nome de usuário e você 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 seguinte configuração 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 or newer have reasonable default configuration so that no server tweaking should be necessary and all what is needed can be configured on the client side.

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 você esteja planejando ter um grande número de usuários simultâneos acessando 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 você 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 você estar recebendo o erro `#1071 - Specified key was too long; max key length is 767 bytes`, atualize sua configuração para incluir as configurações `innodb` acima e reinicie sua instalação.

Dica: No caso de você estar recebendo 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": {},
    }
}
```

Você também deve criar a conta de usuário `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 ativação de conta e sobre várias notificações configuradas pelos usuários. Para isso, ele precisa de acesso a um servidor SMTP.

A configuração do servidor de e-mail é configurada usando essas configurações: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_SSL`, `EMAIL_USE_TLS`, `EMAIL_HOST_USER` e `EMAIL_PORT`. Seus nomes são bastante autoexplicativos, mas você pode encontrar mais informações na documentação do Django.

Dica: Caso você 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.

Ver também:

Não recebendo e-mails do Weblate, Configurando e-mail de saída em contêiner Docker

Executando por trás de um proxy reverso

Vários recursos no Weblate dependem de ser capaz de obter endereço IP do cliente. Isso inclui *Limitação de taxa*, *Proteção contra spam* ou *Registro de auditoria*.

Na configuração padrão, o Weblate analisa o endereço IP de `REMOTE_ADDR` que é definido pelo manipulador WSGI.

No caso de você estiver usando um proxy reverso, este campo provavelmente conterá seu endereço. Você precisa configurar o Weblate para confiar em cabeçalhos HTTP adicionais e analisar o endereço IP destes. Isso não pode ser ativado por padrão, pois permitiria falsificação de endereço IP para instalações que não usam um proxy reverso. Habilitar `IP_BEHIND_REVERSE_PROXY` pode ser suficiente para as configurações mais usuais, mas você pode precisar ajustar `IP_PROXY_HEADER` e `IP_PROXY_OFFSET` também.

Ver também:

Proteção contra spam, *Limitação de taxa*, *Registro de auditoria*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

Proxy HTTP

O Weblate executa comandos VCS e aqueles 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"
```

Ver também:

Variáveis de ambiente de proxy

2.1.7 Ajustando a configuração

Ver também:

Amostra de configuração

Copie `weblate/settings_example.py` para `weblate/settings.py` e ajuste-o para corresponder à configuração. Você provavelmente vai querer ajustar as seguintes opções: `ADMINS`

Lista de administradores de sites para receber notificações quando algo dá errado, por exemplo, notificações em mensagens fracassadas ou erros de Django.

Ver também:

`ADMINS`

`ALLOWED_HOSTS`

Você precisa definir isso para listar os hosts que seu site deve servir. Por exemplo:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternativamente, você pode incluir curinga:

```
ALLOWED_HOSTS = ["*"]
```

Ver também:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Configuração de hosts permitidos*

`SESSION_ENGINE`

Configure como suas sessões serão armazenadas. Caso você mantenha o mecanismo de backend do banco de dados padrão, você deve agendar: **weblate clearsessions** para remover dados de sessão obsoletos do banco de dados.

Se você estiver usando o Redis como cache (veja [Habilitar o cache](#)) é recomendado usá-lo para sessões também:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Ver também:

[Configuring the session engine](#), `SESSION_ENGINE`

DATABASES

Conectividade ao servidor de banco de dados, verifique a documentação do Django para obter mais detalhes.

Ver também:

[Configuração de banco de dados para o Weblate](#), `DATABASES`, Banco de Dados

DEBUG

Desabilite isso para qualquer servidor de produção. Com o modo depuração ativado, o Django mostrará backtraces em caso de erro aos usuários, quando você desabilitá-lo, erros serão enviados por e-mail para ADMINS (veja acima).

O modo depuração também diminui o Weblate, já que o Django armazena muito mais informações internamente neste caso.

Ver também:

`DEBUG`

DEFAULT_FROM_EMAIL

Endereço de remetente de e-mail para e-mail de saída, por exemplo, e-mails de registro.

Ver também:

`DEFAULT_FROM_EMAIL`

SECRET_KEY

Chave usada por Django para assinar algumas informações em cookies, consulte [Chave secreta do Django](#) para obter mais informações.

Ver também:

`SECRET_KEY`

SERVER_EMAIL

E-mail usado como endereço de remetente para envio de e-mails ao administrador, por exemplo, notificações em mensagens fracassadas.

Ver também:

`SERVER_EMAIL`

2.1.8 Preenchendo o banco de dados

Depois que sua configuração estiver pronta, você pode executar `weblate migrate` para criar a estrutura do banco de dados. Agora você deve ser capaz de criar projetos de tradução usando a interface administrativa.

Caso você queira executar uma instalação não interativamente, você pode usar `weblate migrate --noinput` e, em seguida, criar um usuário administrativo usando `createadmin` comando.

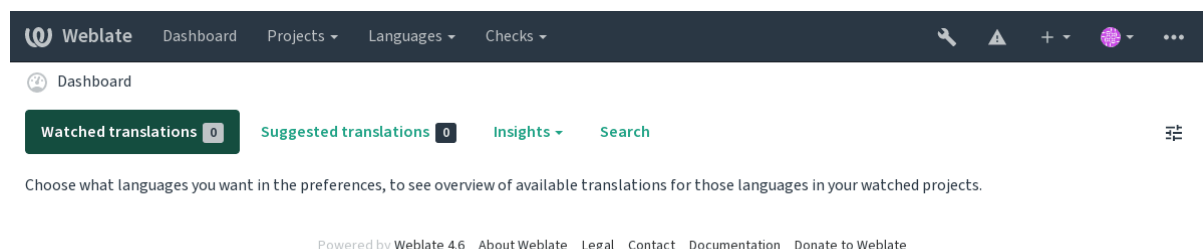
Uma vez feito, você também deve verificar o *Relatório de desempenho* na interface administrativa, o que lhe dará dicas de configuração potencial não ideal em seu site.

Ver também:

[Configuração](#), [Lista de privilégios](#)

2.1.9 Configuração de produção

Para uma configuração de produção, você 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 conectado como um superusuário:



Também é recomendado inspecionar verificações desencadeadas por Django (embora você possa não precisar corrigir todas elas):

```
weblate check --deploy
```

Você também pode revisar a mesma lista de verificação na [Interface de gerenciamento](#).

Ver também:

[Lista de verificação para distribuição](#)

Desabilitar o modo de depuração

Desabilite o modo de depuração do Django (`DEBUG`) com:

```
DEBUG = False
```

Com o modo depuração ativado, o Django armazena todas as consultas executadas e mostra aos usuários atrasos de erros, o que não é desejado em uma configuração de produção.

Ver também:

[Ajustando a configuração](#)

Configurar corretamente administradores

Defina os endereços de administração corretos para a configuração `ADMINS` para definir quem receberá e-mails caso algo dê errado no servidor, por exemplo:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Ver também:

Ajustando a configuração

Definir domínio correto do site

Ajuste o nome e o domínio do site na interface administrativa, caso contrário, links no RSS ou e-mails de registro não funcionarão. Isso 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](#).

Ver também:

Configuração de hosts permitidos, Configurar corretamente HTTPS `SITE_DOMAIN`, `WEBLATE_SITE_DOMAIN`, `ENABLE_HTTPS`

Configurar corretamente HTTPS

É fortemente recomendado executar Weblate usando o protocolo HTTPS criptografado. Depois de habilitá-lo, você deve definir `ENABLE_HTTPS` nas configurações:

```
ENABLE_HTTPS = True
```

Dica: Você pode querer configurar o HSTS também, consulte [SSL/HTTPS](#) para obter mais detalhes.

Ver também:

`ENABLE_HTTPS`, Configuração de hosts permitidos, Definir domínio correto do site

Definir corretamente `SECURE_HSTS_SECONDS`

Se o seu site for servido sobre SSL, você deve considerar definir um valor para **:django: `SECURE_HSTS_SECONDS`** no `settings.py` para habilitar HTTP Strict Transport Security. Por padrão, ele está definido para 0 como mostrado abaixo.

```
SECURE_HSTS_SECONDS = 0
```

Se 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 isso incorretamente pode quebrar irreversivelmente (por algum tempo) 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.

Ver também:

Configuração de banco de dados para o Weblate, *Migrando de outros bancos de dados para o PostgreSQL*, *Ajustando a configuração*, Banco de Dados

Habilitar o cache

Se possível, use Redis do Django ajustando 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",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
}
```

Dica: Caso você altere as configurações do Redis para o cache, pode ser necessário ajustá-las também para o Celery, consulte *Tarefas de fundo usando Celery*.

Ver também:

Cache de avatares, O framework de “cache” do Django

Cache de avatares

Além do cache de Django, Weblate realiza cache de avatares. Recomenda-se usar um cache separado, baseado em arquivos 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,
        },
    },
}
```

Ver também:

[ENABLE_AVATARS](#), [AVATAR_URL_PREFIX](#), [Avatars](#), [Habilitar o cache](#), O framework de “cache” do Django

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 `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 desabilitar o envio de e-mails pelo Weblate, defina `EMAIL_BACKEND` para `django.core.mail.backends.dummy.EmailBackend`.

Isso desabilitará *toda* entrega de e-mail, incluindo e-mails de registro ou redefinição de senha.

Ver também:

[Ajustando 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 seu site pode servir, deixando-o vazio bloqueará quaisquer solicitações.

Caso isso não esteja configurado para corresponder ao seu servidor HTTP, você terá erros como Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.

Dica: No contêiner Docker, isso está disponível como `WEBLATE_ALLOWED_HOSTS`.

Ver 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 seu próprio valor em vez de usar o da configuração do exemplo.

Você pode gerar uma nova chave usando `weblate/examples/generate-secret-key`, que vem com o Weblate.

Ver também:

[SECRET_KEY](#)

Diretório inicial

Alterado na versão 2.1: Isso não é mais necessário, o Weblate agora armazena todos os seus dados em `DATA_DIR`.

O diretório home do usuário que executa o Weblate deve existir e ser escrito por este usuário. Isso é especialmente necessário se você quiser usar o SSH para acessar repositórios privados, mas o Git pode precisar acessar este diretório também (dependendo da versão git que você usa).

Você 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 para o diretório home do usuário é definido em `/etc/passwd`. Muitas distribuições usam como padrão um diretório sem permissão de escrita para usuários usados para servir conteúdo web (como `apache`, `www-data` ou `wwwrun`), então você tem que executar o Weblate sob um usuário diferente ou alterar essa configuração.

Ver também:

Acessando repositórios

Carregamento de modelos

Recomenda-se usar um carregador de modelo em cache para Django. Ele armazena modelos analisados e evita a necessidade de fazer análise a cada solicitação. Você pode configurá-lo usando 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",
                    ],
                ),
            ],
        },
    ],
]
```

Ver também:

`django.template.loaders.cached.Loader`

Executando 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 usando Celery* e cobre 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 (diariamente).
- 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 padrão de executar essas tarefas é usar o Celery e o Weblate já vem com a configuração adequada, consulte *Tarefas de fundo usando 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 padrão. Caso não seja o caso em seu sistema, altere as localidades para a variante UTF-8.

Por exemplo, editando `/etc/default/locale` e definindo lá `LANG="C.UTF-8"`.

Em alguns casos, os serviços individuais têm configuração separada para locais. Por exemplo, ao usar o Apache, você pode querer defini-lo em `/etc/apache2/envvars`:

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

Usando autoridade certificadora personalizada

O Weblate verifica os certificados SSL durante as solicitações HTTP. Caso você esteja usando uma autoridade de certificação personalizada que não seja confiável em pacotes padrão, você terá que adicionar 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 colocando o certificado de 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 Python, você precisará configurar solicitações para usar o pacote de AC do sistema em vez do enviado com ele. Isso pode ser conseguido colocando seguintes trechos para `settings.py` (o caminho é específico do Debian):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Comprimindo os ativos do cliente

Weblate vem com um monte de arquivos JavaScript e CSS. Por razões de desempenho, é bom comprimi-los antes de enviar para um cliente. Na configuração padrão isso é feito na mosca 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 habilitar `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 você precisa compactar os arquivos para corresponder à versão atual:

```
weblate compress
```

Dica: A imagem oficial do Docker já tem este recurso habilitado.

Ver também:

[Common Deployment Scenarios](#), [Servindo arquivos estáticos](#)

2.1.10 Executando servidor

Dica: No caso de você não ter experiência com os serviços descritos abaixo, você pode tentar seguir [Instalando usando Docker](#).

Você 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 [Habilitar o cache](#))
- Servidor web frontend para arquivos estáticos e terminação SSL (consulte [Servindo arquivos estáticos](#))
- Servidor WSGI para conteúdo dinâmico (consulte [Amostra de configuração para NGINX e uWSGI](#))
- Celery para executar tarefas em segundo plano (consulte [Tarefas de fundo usando 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, você executará todos os serviços em um único servidor (virtual), mas no caso de sua instalação estar muito carregada, você 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 usuário que o processo do Celery, caso contrário, os arquivos em `DATA_DIR` serão armazenados com propriedade mista, levando a problemas de tempo de execução.

Veja também [Permissões do sistema de arquivos](#) e [Tarefas de fundo usando Celery](#).

Executando 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, você pode usar o servidor web embutido no Django:

```
weblate runserver
```

Aviso: NÃO USE ESTE SERVIDOR EM UMA CONFIGURAÇÃO DE PRODUÇÃO. Ele 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 arquivos estáticos com `DEBUG` ativado, pois é destinado apenas ao desenvolvimento. Para uso da produção, consulte as configurações de wsgi em *Amostra de configuração para NGINX e uWSGI*, *Amostra de configuração para o Apache*, *Amostra de configuração para Apache e Gunicorn* e *Servindo arquivos estáticos*.

Servindo arquivos estáticos

Alterado na versão 2.4: Antes da versão 2.4, o Weblate não usava corretamente a estrutura de arquivos estáticos do Django e a configuração era mais complexa.

Django precisa coletar seus arquivos estáticos em um único diretório. Para isso, execute `weblate collectstatic --noinput`. Isso copiará os arquivos estáticos em um diretório especificado pela configuração `STATIC_ROOT` (isso é padrão para um diretório `static` dentro de `DATA_DIR`).

Recomenda-se servir arquivos estáticos diretamente do seu servidor web. Você deve usá-los para os seguintes caminhos:

/static/ Serve arquivos estáticos para Weblate e a interface de administração (definida por `STATIC_ROOT`).

/media/ Usado para envio de mídia pelo usuário (por exemplo, capturas de tela).

/favicon.ico Deve ser reescrito para reescrever uma regra para servir `/static/favicon.ico`.

Ver também:

Amostra de configuração para NGINX e uWSGI, *Amostra de configuração para o Apache*, *Amostra de configuração para Apache e Gunicorn*, *Comprimindo os ativos do cliente*, *Fazendo deploy do Django*, *Deploy de arquivos estáticos*

Política de segurança de conteúdo

A configuração padrão do Weblate habilita o middleware `weblate.middleware.SecurityMiddleware` que define cabeçalhos HTTP relacionados à segurança como `Content-Security-Policy` ou `X-XSS-Protection`. Eles são configurados por padrão para funcionar com o Weblate e sua configuração, mas isso pode precisar de personalização para o seu ambiente.

Ver também:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

Amostra de configuração 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`). Não se esqueça de definir o caminho de pesquisa Python para seu virtualenv também (por exemplo, usando `virtualenv = /home/user/weblate-env`` no uWSGI).

A configuração a seguir executa o Weblate como uWSGI sob o servidor web NGINX.

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/;
        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
```

(continua na próxima página)

(continuação da página anterior)

```
# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path = /path/to/weblate

# In case you're using virtualenv uncomment this:
virtualenv = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size = 8192

# Reload when consuming too much of memory
reload-on-rss = 250

# Increase number of workers for heavily loaded sites
workers = 8

# Enable threads for Sentry error submission
enable-threads = true

# Child processes do not need file descriptors
close-on-exec = true

# Avoid default 0000 umask
umask = 0022

# Run as weblate user
uid = weblate
gid = weblate

# Enable harakiri mode (kill requests after some time)
# harakiri = 3600
# harakiri-verbose = true

# Enable uWSGI stats server
# stats = :1717
# stats-http = true

# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

Ver também:

Como usar Django com uWSGI

Amostra de configuração para o Apache

Recomenda-se o uso de prefork de MPM ao usar WSGI com Weblate.

A configuração a seguir executa o Weblate como WSGI. Você precisa ter habilitado `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.
#
```

(continua na próxima página)

(continuação da página anterior)

```

<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

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

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

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

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

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

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

```

Nota: Weblate precisa do Python 3, então, por favor se certifique que você está executando a variante do Python 3 do modwsgi. Usualmente, está disponível como um pacote separado, por exemplo `libapache2-mod-wsgi-py3`.

Ver também:

Codificação e localidades do sistema, Como usar o Django com Apache e mod_wsgi

Amostra de configuração para Apache e Gunicorn

A configuração a seguir executa o Weblate em Gunicorn e 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

```

(continua na próxima página)

(continuação da página anterior)

```

ServerName weblate.example.org

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

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

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

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

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

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

```

Ver também:

[tutorial/deployment/wsgi/gunicorn](#)

Rodando 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, usando `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/

```

(continua na próxima página)

(continuação da página anterior)

```

<Directory /home/weblate/data/static/>
    Require all granted
</Directory>

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

# Path to your Weblate virtualenv
WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate
WSGIProcessGroup weblate
WSGIApplicationGroup %{GLOBAL}

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

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

</VirtualHost>

```

Adicionalmente, você irá ter de ajustar o `weblate/settings.py`:

```
URL_PREFIX = "/weblate"
```

2.1.11 Tarefas de fundo usando Celery

Novo na versão 3.2.

O Weblate usa Celery para processar tarefas em segundo plano. Uma configuração típica usando redis como backend se parece com isso:

```

CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

```

Ver também:

Configuração do broker Redis no Celery

Para o desenvolvimento, você pode querer 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

```

Você 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 usuário que o Weblate e o processo do WSGI, caso contrário, os arquivos em `DATA_DIR` serão armazenados com propriedade mista, levando a problemas de tempo de execução.

Veja também *Permissões do sistema de arquivos* e *Executando servidor*.

Executando Celery como serviço do sistema

Provavelmente você vai querer executar o Celery como um daemon e que é coberto por [Daemonization](#). Para as configurações Linux mais comuns usando systemd, você pode usar os arquivos de exemplo disponibilizados na pasta `examples` listado abaixo.

Unidade do systemd a ser colocada 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} \
  -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 colocada 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"
```

(continua na próxima página)

(continuação da página anterior)

```
# 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 usando **logrotate** a ser colocada como `/etc/logrotate.d/celery`:

```
/var/log/celery/*.log {
    weekly
    missingok
    rotate 12
    compress
    notifempty
}
```

Tarefas periódicas usando a batida do Celery

O Weblate vem com configuração embutida para tarefas programadas. No entanto, você 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 funcionando corretamente, pode não estar funcionando ou seu banco de dados foi corrompido. Verifique os registros de inicialização do Celery nesse caso para descobrir a causa raiz.

Monitorando status do Celery

Você pode usar `celery_queues` para ver o comprimento atual das filas de tarefas do Celery. Caso a fila fique muito longa, você 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 usuário. Caso você queira ter uma visão geral sobre tais falhas, recomenda-se ajustar a configuração para ir [Coletando relatórios de erros](#).

Ver 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.

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 você 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, você 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 você 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 você 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, você pode ter várias opções para migrar o banco de dados. O mais simples é despejar o banco de dados em um servidor e importá-lo no novo. Alternativamente, você pode usar a replicação no caso de 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 você quiser migrar entre diferentes bancos de dados, a única opção pode ser usar o gerenciamento 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. Você 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 em sua nuvem. Encontre um guia detalhado para sua plataforma:

- *Instalando usando 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 equipe 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 aplicativos da web. Weblate instalado com Cloudron será automaticamente atualizado. O pacote é mantido pela equipe Cloudron em seu *repositório de pacotes Weblate* <<https://git.cloudron.io/cloudron/weblate-app>> _.



Weblate no YunoHost

O projeto de auto-hospedagem YunoHost fornece um pacote para Weblate. Uma vez que você tenha a sua instalação YunoHost, você pode instalar o Weblate como qualquer outro aplicativo. Ele fornecerá uma pilha de trabalho completo com backup e restauração, mas você ainda pode ter que editar seu arquivo de configurações para usos específicos.

Você pode usar sua interface de administração ou este botão (ele 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 usando Docker*) possui todas as etapas de atualização integradas. Não há nenhuma etapa manual além de baixar 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 *Fazendo backup e movendo 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 *Executando servidor* and *Servindo arquivos estáticos*):

```
weblate collectstatic --noinput
```

6. Compactar arquivos JavaScript e CSS (opcional, veja *Comprimindo 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 usando 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.

Ver 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.

Ver 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 *DeepL* machine translation now defaults to v2 API, you might need to adjust `MT_DEEPL_API_VERSION` in case your current DeepL subscription does not support that.

Ver 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.

Ver 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 *Weblate Client* 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.

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

Ver 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.

Ver 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.

Ver 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 migrate to PostgreSQL as that will be the only supported database backend in the 4.0 release. The following steps will guide you in migrating your data between the databases. Please remember to stop both web and Celery servers prior to the migration, otherwise you might end up with inconsistent data.

Criando um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate em um banco de dados separado e separar a conta do usuário:

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

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

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

Migrating using Django JSON dumps

The simplest approach for migration is to utilize Django JSON dumps. This works well for smaller installations. On bigger sites you might want to use pgloader instead, see [Migrating to PostgreSQL using pgloader](#).

1. Add PostgreSQL as additional database connection to the `settings.py`:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
```

(continua na próxima página)

```

    "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 Fazendo backup e movendo o Weblate

2.4.1 Backup automatizado usando BorgBackup

Novo na versão 3.9.

O Weblate tem suporte embutido para criação de backups de serviços usando [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 gerenciamento da aba *Backups*.

Alterado na versão 4.4.1: Both PostgreSQL and MySQL/MariaDB databases are included in the automated backups.

Os backups que usam o Borg são incrementais e o Weblate é configurado para manter os seguintes backups:

- Backups diários para 14 dias
- Weekly backups for 8 weeks back
- Monthly backups for 6 months back

Weblate
Dashboard
Projects
Languages
Checks

+

Manage / Backups

Backup process triggered

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

Backup service: /tmp/tmpkrnjny6weblate

Backup service credentials
April 18, 2021

Backup repository
/tmp/tmpkrnjny6weblate

Passphrase
ZbTHOQCyWyE085zWc1xR1G(pr*UaPFg\$TDhVMYfbpBBCj)0A2g
The passphrase is used to encrypt the backups and is necessary to restore them.

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

Deleted the oldest backups
April 18, 2021

Backup performed
April 18, 2021

Repository initialization
April 18, 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 4.6
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Chave de criptografia do Borg

BorgBackup cria backups criptografados e você não conseguiria restaurá-los sem a senha. A senha é gerada ao adicionar novo serviço de backup e você deve copiá-lo e mantê-lo em um lugar seguro.

Se você estiver usando *Armazenamento de backup provisionado do Weblate*, faça backup da sua chave SSH privada também, pois ela é usada para acessar seus backups.

Ver também:

`borg init`

2.4.2 Armazenamento de backup provisionado do Weblate

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

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

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

2.4.3 Usando armazenamento de backup personalizado

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

Ver também:

[General](#) na documentação do Borg

Sistema de arquivos local

It is recommended to specify the absolute path for the local backup, for example `/path/to/backup`. The directory has to be writable by the user running Weblate (see *Permissões do sistema de arquivos*). If it doesn't exist, Weblate attempts to create it but needs the appropriate permissions to do so.

Dica: When running Weblate in Docker, please ensure the backup location is exposed as a volume from the Weblate container. Otherwise the backups will be discarded by Docker upon restarting the container it is in.

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

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

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

The directory where backups will be stored have to be owned by UID 1000, otherwise Weblate won't be able to write the backups there.

Backups remotos

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

Dica: *Armazenamento de backup provisionado do Weblate* fornece backups remotos automatizados.

Ver também:

Chave SSH do Weblate

2.4.4 Restaurando do BorgBackup

1. Restaure o acesso ao repositório de backup e prepare sua senha de backup.
2. List all the backups on the server using `borg list REPOSITORY`.
3. Restore the desired backup to the current directory using `borg extract REPOSITORY::ARCHIVE`.
4. Restaure o banco de dados a partir do despejo de SQL colocado no diretório `backup` no diretório de dados do Weblate (veja *Dados despejados para os backups*).
5. Copy the Weblate configuration (`backups/settings.py`, see *Dados despejados para os backups*) to the correct location, see *Ajustando a configuração*.
6. Copy the whole restored data dir to the location configured by `DATA_DIR`.

The Borg session might look like this:

```
$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
2019-09-26T14:56:08                               Thu, 2019-09-26 14:56:08
→ [de0e0f13643635d5090e9896bdaceb92a023050749ad3f3350e788f1a65576a5]
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:
```

Ver também:

`borg list`, `borg extract`

2.4.5 Backup manual

Depending on what you want to save, back up the type of data Weblate stores in each respective place.

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

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

Banco de dados

O local de armazenamento real depende da configuração do seu banco de dados.

Dica: The database is the most important storage. Set up regular backups of your database. Without the database, all the translations are gone.

Backup nativo do banco de dados

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

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

Backup do banco de dados do Django

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

Prior to restoring the database you need to be running exactly the same Weblate version the backup was made on. This is necessary as the database structure does change between releases and you would end up corrupting the data in some way. After installing the same version, run all database migrations using `migrate`.

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

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

Arquivos

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

Dados despejados para os backups

Armazenados em `DATA_DIR/backups`.

O Weblate despeja vários dados aqui, e você pode incluir esses arquivos para backups mais completos. Os arquivos são atualizados diariamente (requer um servidor de “beats” do Celery em execução, consulte *Tarefas de fundo usando Celery*). Atualmente, isso 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`.

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

Repositórios de controle de versão

Armazenados em `DATA_DIR/vcs`.

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

Chaves SSH e GPG

Armazenados em `DATA_DIR/ssh` e `DATA_DIR/home`.

If you are using SSH or GPG keys generated by Weblate, you should back up these locations. Otherwise you will lose the private keys and you will have to regenerate new ones.

Arquivos enviados pelo usuário

Armazenados em `DATA_DIR/media`.

You should back up all user uploaded files (e.g. *Contexto visual para textos*).

Tarefas do Celery

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

Ver também:

Tarefas de fundo usando Celery

Linha de comando para backup manual

Using a cron job, you can set up a Bash command to be executed on a daily basis, for example:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz \
↪backups vcs ssh home media fonts secret
```

The string between the quotes after `XZ_OPT` allows you to choose your xz options, for instance the amount of memory used for compression; see <https://linux.die.net/man/1/xz>

You can adjust the list of folders and files to your needs. To avoid saving the translation memory (in backups folder), you can use:

```
$ 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 Restaurando backup manual

1. Restaure todos os dados dos quais você tenha feito backup.
2. Atualize todos repositórios usando o `updategit`.

```
weblate updategit --all
```

2.4.7 Movendo uma instalação do Weblate

Relocate your installation to a different system by following the backing up and restoration instructions above.

Ver também:

Upgrading from Python 2 to Python 3, Migrando de outros bancos de dados para o PostgreSQL

2.5 Autenticação

2.5.1 Registro de usuário

A configuração padrão para Weblate é usar python-social-auth, um formulário no site para lidar com o registro de novos usuários. Depois de confirmar seu e-mail, um novo usuário pode contribuir ou autenticar usando um dos serviços de terceiros.

Você também pode desativar o registro de novos usuários usando `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 o fazer. Utilizando-a, você pode importar o banco de dados de usuários de outros projetos baseados no Django (veja *Migrating from Pootle*).

Django pode, adicionalmente, ser configurado para autenticar em outros meios também.

Ver 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 sua documentação para as instruções de configuração genéricas em [Django Framework](#).

Nota: Por padrão, o Weblate conta com serviços de autenticação de terceiros para fornecer um endereço de e-mail validado. Se alguns dos serviços que você deseja usar não suportarem isso, por favor, aplique a validação de e-mail no lado Weblate configurando `FORCE_EMAIL_VALIDATION` para eles. Por exemplo:

```
SOCIAL_AUTH_OPENUSE_FORCE_EMAIL_VALIDATION = True
```

Ver 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 usuário por padrão, você tem que solicitá-lo explicitamente, caso contrário o Weblate não será capaz de pagar corretamente as contribuições que os usuários fazem.

Ver também:

[Backend de Python Social Auth](#)

Autenticação por OpenID

Para serviços baseados em OpenID, geralmente é apenas uma questão de habilitá-los. A seção 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",
)
```

Ver também:

[OpenID](#)

Autenticação por GitHub

Você precisa registrar um aplicativo 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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[GitHub](#)

Autenticação por Bitbucket

Você precisa registrar um aplicativo no Bitbucket e, em seguida, dizer ao Weblate todos os seus segredos:

```
# 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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[Bitbucket](#)

OAuth 2 do Google

Para usar o OAuth 2 do Google, você precisa se registrar em um aplicativo 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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[Google](#)

OAuth 2 do Facebook

Como de costume com os serviços OAuth 2, você precisa registrar seu aplicativo no Facebook. Uma vez feito isso, você 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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

Facebook

OAuth 2 do GitLab

Para usar o OAuth 2 do GitLab, você precisa registrar um aplicativo em `<https://gitlab.com/profile/applications>`.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/gitlab/` e garantir que você 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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

GitLab

Active Directory do Microsoft Azure

Weblate pode ser configurado para usar inquilinos comuns ou específicos para autenticação.

A 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
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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

Microsoft Azure Active Directory

Slack

Para usar o OAuth 2 do Slack, você precisa cadastrar um aplicativo 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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[Slack](#)

Desativando autenticação por senha

Autenticação por e-mail e senha pode ser desativada através da remoção de `social_core.backends.email.EmailAuth` de `AUTHENTICATION_BACKENDS`. Mantenha sempre `weblate.accounts.auth.WeblateUserBackend` lá, ele é necessário para a funcionalidade central do Weblate.

Dica: Você ainda pode usar autenticação por senha para a interface administrativa, para usuários que você cria manualmente lá. Basta navegar para `/admin/`.

Por exemplo, a autenticação usando apenas o provedor Open ID do openSUSE pode ser alcançada usando o seguinte:

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

2.5.4 Autenticação por senha

O `settings.py` padrão vem com um razoável conjunto de `AUTH_PASSWORD_VALIDATORS`:

- As senhas não podem ser muito similares às suas outras informações pessoais.
- As senhas devem conter no mínimo 10 caracteres.
- As senhas não podem ser uma senha comumente usada.
- As senhas não podem ser inteiramente numéricas.
- As senhas não podem consistir em um único caractere ou apenas espaço em branco.
- As senhas não podem corresponder a uma senha que você usou no passado.

Você pode personalizar esta configuração para corresponder à sua política de senha.

Além disso, você também pode instalar o [django-zxcvbn-password](#) o que dá bastante estimativas realistas de senha dificuldade e permite rejeitar senhas 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
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbn1+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
```

Ver 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*. Você 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
```

Aviso: Com *django-auth-ldap* anterior a 1.3.0, o *Associações automáticas de grupo* não funcionarão corretamente para usuários recém-criados.

Nota: Há algumas incompatibilidades no módulo Python LDAP 3.1.0, o que pode impedir você de usar essa versão. Se você obter o erro `AttributeError: 'module' object has no attribute '_trace_level'`, fazer o downgrade para *python-ldap* 3.0.0 pode ajudar.

Uma vez que você tenha o pacote instalado, você pode conectá-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"
```

(continua na próxima página)

(continuação da página anterior)

```
# 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: Você deve remover 'social_core.backends.email.EmailAuth' da configuração `AUTHENTICATION_BACKENDS`; caso contrário, os usuários poderão definir sua senha no Weblate e autenticar usando isso. Manter 'weblate.accounts.auth.WeblateUserBackend' ainda é necessário para fazer permissões e facilitar usuários anônimos. Ele também permitirá que você faça login usando uma conta administrativa local, se você a criou (por exemplo, usando `createadmin`).

Usando senha associada

Se você não puder usar associação direta para autenticação, você precisará usar a pesquisa e fornecer um usuário 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 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
```

(continua na próxima página)

(continuação da página anterior)

```

AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "OU=Groups,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True

```

Ver também:

Django Authentication Using LDAP, Authentication

2.5.7 Autenticação por CAS

A autenticação por CAS pode ser alcançada usando um pacote como o *django-cas-ng*.

O primeiro passo é divulgar o campo de e-mail do usuário via CAS. Isso tem que ser configurado no próprio servidor CAS, e requer que você execute 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 instalado, você pode conectá-lo ao sistema de autenticação do Django modificando o arquivo `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 usuário. Para que isso funcione, você tem que importar o sinal do pacote *django-cas-ng* e conectar seu código com este sinal. Fazer isso em configurações de arquivo pode causar problemas, portanto, é sugerido colocá-lo:

- No método `django.apps.AppConfig.ready()` da configuração do seu aplicativo
- No arquivo `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)

```

(continua na próxima página)

(continuação da página anterior)

```
def update_user_email_address(sender, user=None, attributes=None, **kwargs):
    # If your CAS server does not always include the email attribute
    # you can wrap the next two lines of code in a try/catch block.
    user.email = attributes["email"]
    user.save()
```

Ver também:

Django CAS NG

2.5.8 Configurando autenticação por Django de terceiros

Geralmente, qualquer plugin de autenticação Django deve funcionar com Weblate. Basta seguir as instruções do plugin, lembrando de manter o backend do usuário Weblate instalado.

Ver também:

Autenticação por LDAP, Autenticação por CAS

Normalmente, a instalação consiste em adicionar uma autenticação de backend a `AUTHENTICATION_BACKENDS` e a instalar um aplicativo 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 Controle de acesso

O Weblate vem com um sistema de privilégios fino para atribuir permissões ao usuário para toda a instância ou em um 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 você está usando.

2.6.1 Controle de acesso personalizado

Se você não está administrando toda a instalação do Weblate e apenas tem acesso para gerenciar certos projetos (como no [Hosted Weblate](#)), suas opções de gerenciamento de controle de acesso são limitadas às seguintes configurações. Se você não precisa de nenhuma configuração complexa, elas são suficientes para você.

Controle de acesso do projeto

Nota: Este recurso está indisponível para os projetos que executam o plano Libre no Hosted Weblate.

Você pode limitar o acesso do usuário a projetos individuais selecionando uma configuração diferente de *Controle de acesso*. As opções disponíveis são:

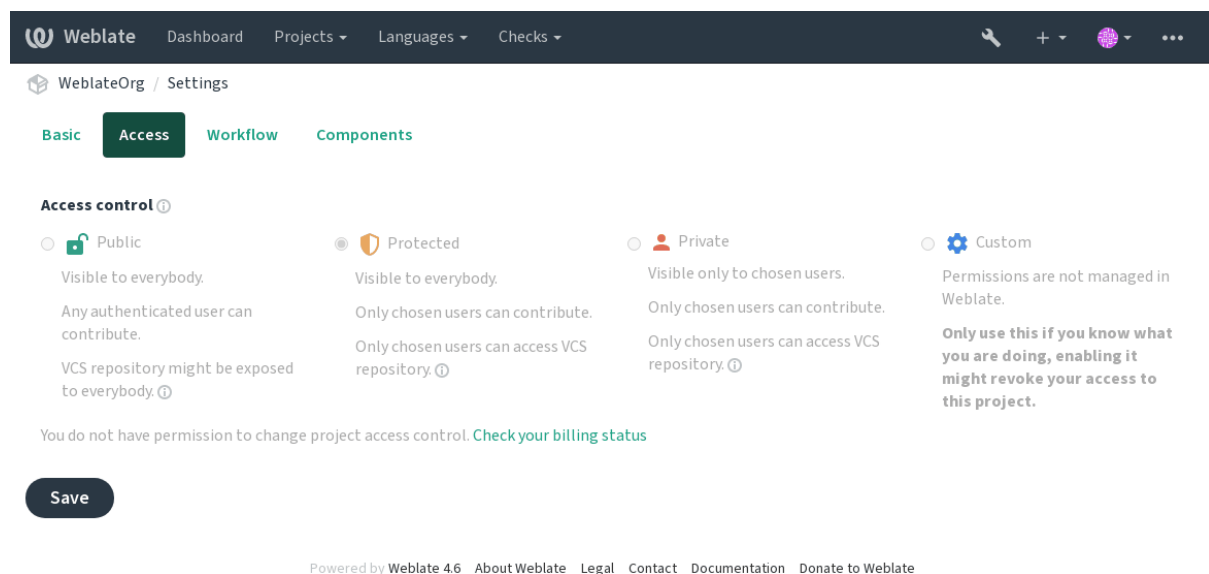
Público Visível publicamente, traduzível para todos os usuários autenticados.

Protegido Publicamente visível, mas somente traduzível por usuários selecionados.

Privado Visível e traduzível apenas por usuários selecionados.

Personalizado Os recursos de *Gerenciamento de usuários* serão desabilitados; por padrão, todos os usuários estão proibidos de realizar qualquer ação no projeto. Você terá que configurar todas as permissões usando *Controle de acesso personalizado*.

O *Controle de acesso* pode ser alterado na aba *Acesso* da configuração (*Gerenciar* ↓ *Configurações*) de cada projeto.



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 usuários por padrão em projetos *Públicos*, *Protegidos* e *Privados* pode ser redefinido pelo administrador da instância de Weblate usando *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 você estiver fazendo isso sem permissão administrativa para toda a instância de Weblate, você perderá instantaneamente o seu acesso para gerenciar o projeto.

Ver também:

[Controle de acesso](#)

Gerenciando controle de acesso por projeto

Os usuários com o privilégio *Gerenciar acesso ao projeto* (veja a [Lista de privilégios](#)) podem gerenciar usuários em projetos com controle de acesso não *Personalizado*. Eles podem atribuir os usuários 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 textos fonte (se permitido nas [configurações do projeto](#)) e informações de textos fonte.

Idiomas Pode gerenciar idiomas traduzidos (adicionar ou remover traduções).

Glossário Pode gerenciar glossário (adicionar ou remover entradas, e também enviar).

Memória Pode gerenciar memória de tradução.

Capturas de tela Pode gerenciar capturas de tela (adicioná-las ou removê-las e associá-las a textos fonte).

VCS Pode gerenciar VCS e acessar o repositório exportado.

Cobrança Pode acessar informações de cobrança e configurações (consulte [Cobrança](#)).

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 usuários.

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 usuário do Weblate.

Users

Username	Full name	E-mail	Last login	Administration	Billing	Glossary	Languages	Memory	Screenshots	Sources	Translate	VCS
testuser	Webplate Test	weblate@example.org	25 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

Invite new user

E-mail

Username

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

Full name

Invite

Powered by Weblate 4.6 [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 *Gerenciar* ↓ *Usuários*.

Novo convite de usuário

Além de adicionar um usuário existente ao projeto, também é possível convidar novos. Qualquer novo usuário será criado imediatamente, mas a conta permanecerá inativa até que você entre com um link no convite enviado por e-mail. Não é necessário ter nenhum privilégio de todo o site para fazer isso, a permissão de gerenciamento de acesso no escopo do projeto (por exemplo, uma associação no grupo *Administração*) seria suficiente.

Dica: Se o usuário convidado perdeu a validade do convite, ele pode definir sua senha usando o endereço de e-mail convidado no formulário de redefinição de senha, pois a conta já foi criada.

Novo na versão 3.11: É possível reenviar o e-mail para convites de usuário (invalidando qualquer convite enviado anteriormente).

O mesmo tipo de convite está disponível em todo o site a partir da *interface de gerenciamento* na aba *Usuários*.

Gerenciamento de permissões por projeto

Você pode definir seus projetos para *Protegido* ou *Privado* e *gerenciar usuários* por projeto na interface de usuário do Weblate.

Por padrão, isso impede que o Weblate conceda acesso fornecido por *grupos padrão* *Usuários* e *Visualizadores* devido à própria configuração desses grupos. Isso não impede que você conceda permissões a esses projetos em todo o site, alterando grupos padrão, criando um novo ou criando configurações personalizadas adicionais para componentes individuais, conforme descrito em *Controle de acesso personalizado* abaixo.

Um dos principais benefícios de gerenciar permissões por meio da interface de usuário do Weblate é que você pode delegá-las a outros usuários sem dar a eles o privilégio de superusuário. 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 para os usuários e traduções. Veja *Usuários, 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*. Você pode usá-la para gerenciar as permissões de qualquer projeto. Você não precisa necessariamente definir o *controle de acesso* para *Personalizado* para utilizá-lo. No entanto, você deve ter privilégios de superusuário para usá-lo.

Se você 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 seção *Controle de acesso personalizado*.

Configurações comuns

Esta seção contém uma visão geral de algumas configurações comuns nas quais você pode estar interessado.

Gerenciamento de permissões para todo o site

Para gerenciar permissões para uma instância inteira de uma vez, adicione usuários aos *grupos padrão*:

- *Usuários* (isso é feito por padrão pela *associação de grupo automática*).
- *Revisores* (se você estiver usando *fluxo de trabalho de revisão* com revisores dedicados).
- *Gestores* (se você deseja delegar a maioria das operações de gestão a outra pessoa).

Você deve manter todos os projetos configurados como *Públicos* (veja *Controle de acesso do projeto*), caso contrário as permissões de todo o site fornecidas pela participação nos grupos *Usuários* e *Revisores* não terão nenhum efeito.

Você também pode conceder algumas permissões adicionais de sua escolha aos grupos padrão. Por exemplo, você pode querer dar permissão para gerenciar capturas de tela para todos os *Usuários*.

Você também pode definir alguns novos grupos personalizados. Se você deseja continuar gerenciando 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

Você pode criar seus próprios grupos dedicados para gerenciar permissões para objetos distintos, como linguagens, componentes e projetos. Embora esses grupos só possam conceder privilégios adicionais, você não pode revogar qualquer permissão concedida por grupos de todo o site ou por projeto adicionando outro grupo personalizado.

Exemplo:

Se você quiser (por qualquer motivo) permitir a tradução para um idioma específico (digamos, *tcheco*) apenas para um conjunto fechado de tradutores confiáveis, mantendo as traduções para outros idiomas públicos, você terá que:

1. Remover a permissão para traduzir *Tcheco* de todos os usuários. Na configuração padrão, isso pode ser feito alterando o *grupo padrão* *Usuários*.

Tabela 1: Grupo *Usuários*

Seleção de idioma	<i>Como definido</i>
Idiomas	Todos, exceto <i>Tcheco</i>

2. Adicionar um grupo dedicado para tradutores(as) de *Tcheco*.

Tabela 2: Grupo *Tradutores(as) de tcheco*

Funções	<i>Usuários(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 usuários aos quais deseja conceder permissões a este grupo.

Como você pode ver, o gerenciamento de permissões dessa maneira é poderoso, mas pode ser um trabalho tedioso. Você não pode delegá-lo a outro usuário, a menos que conceda permissões de superusuário.

Usuários, papéis, grupos e permissões

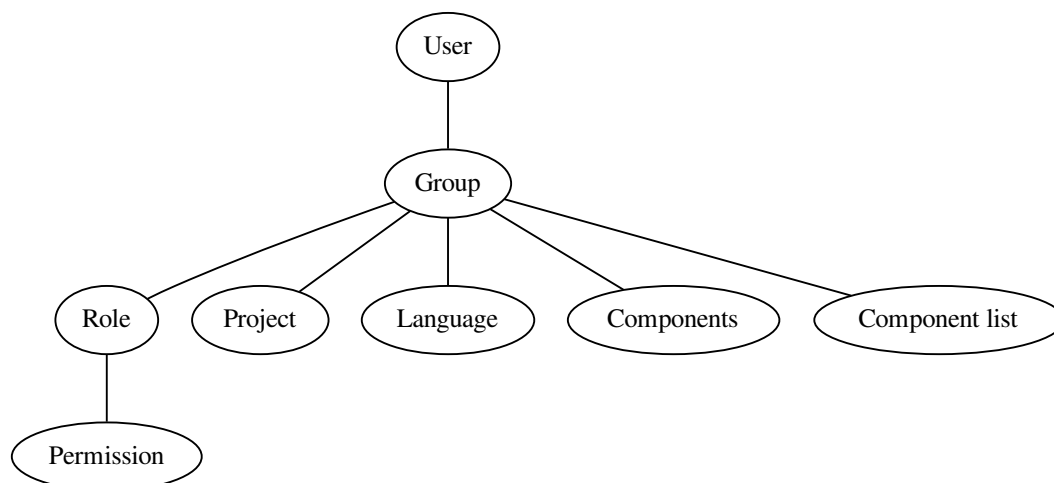
Os modelos de autenticação consistem em vários objetos:

Permissão Permissão individual definida pelo Weblate. Permissões não podem ser atribuídas a usuários. Isso só pode ser feito através da atribuição de papéis.

Papel Um papel define um conjunto de permissões. Isso permite a reutilização desses conjuntos em vários lugares, facilitando a administração.

Usuário Usuário pode pertencer a vários grupos.

Grupo Grupo conecta papéis, usuários e objetos de autenticação (projetos, idiomas e listas de componentes).



Nota: Um grupo não pode ter papéis atribuídos a ele; nesse caso, o acesso para navegar no projeto por qualquer pessoa é presumido (veja abaixo).

Acesso para navegar até um projeto

Um usuário 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 usuário pode acessar componentes irrestritos assim que for possível acessar o projeto dos componentes (e terá todas as permissões que foram concedidas ao usuário 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.
- Otherwise, if the group specifies any *Projects*, either by directly listing them or by having *Projects selection* set to a value like *All public projects*, all those permissions are applied to all the projects, which effectively grants the same permissions to access all projects *unrestricted components*.

- The restrictions imposed by a group's *Languages* are applied separately, when it's verified if a user has an access to perform certain actions. Namely, it's applied only to actions directly related to the translation process itself like reviewing, saving translations, adding suggestions, etc.

Dica: Use *Seleção de idioma* ou *Seleção de projeto* para automatizar a inclusão de todos os idiomas ou projetos.

Exemplo:

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

Tabela 3: Group *Spanish Admin-Reviewers*

Funções	<i>Revisar textos, Gerenciar repositório</i>
Componentes	<code>foo/bar</code>
Idiomas	<i>Espanhol</i>

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

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

Associações automáticas de grupo

Na parte inferior da página de edição *Grupo* na *interface administrativa do Django*, você pode especificar *Associações automáticas de grupo*, que é uma lista de expressões regulares usadas para atribuir automaticamente usuários recém-criados a um grupo com base em seus endereços de e-mail. Esta associação só acontece após a criação da conta.

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

Nota: Associação automática de grupo para *Usuários* e *Visualizadores* sempre será recriada ao atualizar de uma versão anterior do Weblate para outra. Se você deseja desativá-los, defina a expressão regular para `^$` (que nunca corresponderá).

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

Grupos e funções padrão

After installation, a default set of groups is created (see [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. Você não pode alterá-las. Defina uma nova função caso você queira definir seu próprio conjunto de permissões.

Lista de privilégios

Cobrança (consulte [Cobrança](#)) Visualizar informações de cobrança [*Administração, Cobrança*]

Alterações Baixar alterações [*Administração*]

Comentários Publicar comentário [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

Excluir comentário [*Administração*]

Componente Editar configurações do componente [*Administração*]

Bloquear componente, evitando traduções [*Administração*]

Glossário Adicionar entrada do glossário [*Administração, Gerenciar glossário, Usuário avançado*]

Editar entrada do glossário [*Administração, Gerenciar glossário, Usuário avançado*]

Excluir entrada do glossário [*Administração, Gerenciar glossário, Usuário avançado*]

Enviar entradas do glossário [*Administração, Gerenciar glossário, Usuário avançado*]

Sugestões automáticas Usar sugestões automáticas [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

Memória de tradução Editar memória de tradução [*Administração, Gerenciar memória de tradução*]

Excluir memória de tradução [*Administração, Gerenciar memória de tradução*]

Projetos Editar configurações do projeto [*Administração*]

Gerenciar acesso ao projeto [*Administração*]

Relatórios Baixar relatórios [*Administração*]

Capturas de tela Adicionar captura de tela [*Administração, Gerenciar capturas de tela*]

Editar captura de tela [*Administração, Gerenciar capturas de tela*]

Excluir captura de tela [*Administração, Gerenciar capturas de tela*]

Textos fonte Editar informações adicionais de texto [*Administração, Editar fonte*]

Textos Adicionar novo texto [*Administração*]

Remover um texto [*Administração*]

Ignorar verificação com falha [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

Editar textos [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

Revisar textos [*Administração, Revisar textos*]

Editar texto quando as sugestões são forçadas [*Administração, Revisar textos*]

Editar textos fonte [*Administração, Editar fonte, Usuário avançado*]

Sugestões Aceitar sugestão [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

Adicionar sugestão [*Administração, Editar fonte, `` Adicionar sugestão, Usuário avançado, Revisar textos, Traduzir*]

Excluir sugestão [*Administração, Usuário avançado*]

Votar em sugestão [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

Traduções Adicionar idioma para tradução [*Administração, Usuário avançado, Gerenciar idiomas*]

Efetuar tradução automática [*Administração, Gerenciar idiomas*]

Excluir tradução existente [*Administração, Gerenciar idiomas*]

Adicionar idiomas para tradução [*Administração, Gerenciar idiomas*]

Envios Definir autor da tradução enviada [*Administração*]

Sobrescrever textos existentes com envio [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

Enviar traduções [*Administração, Editar fonte, Usuário avançado, Revisar textos, Traduzir*]

VCS Acessar o repositório interno [*Administração, Acessar repositório, Usuário avançado, Gerenciar repositório*]

Fazer commit das alterações para o repositório interno [*Administração, Gerenciar repositório*]

Fazer push das alterações do repositório interno [*Administração, Gerenciar repositório*]

Redefinir as alterações no repositório interno [*Administração, Gerenciar repositório*]

Ver o local do repositório upstream [*Administração, Acessar repositório, Usuário avançado, Gerenciar repositório*]

Atualizar o repositório interno [*Administração, Gerenciar repositório*]

Privilégios para todo o site Usar interface de gerenciamento

adicionar novos projetos

Adicionar definições de idioma

Gerenciar definições de idiomas

Gerenciar grupos

Gerenciar usuários

Gerenciar funções

Gerenciar anúncios

Gerenciar memória de tradução

Gerenciar listas de componentes

Nota: Privilégios para todo o site não são concedidos a nenhuma função padrão. Eles são poderosos e muito próximos do status de superusuário. A maioria deles afetam todos os projetos de sua instalação do Weblate.

Lista de grupos

Os seguintes grupos são criados após a instalação (ou após a execução de `setupgroups`) e você está livre para modificá-los. A migração irá, no entanto, recriá-los se você excluí-los ou renomeá-los.

Convidados Define permissões para usuários não autenticados.

Este grupo contém apenas usuários anônimos (consulte `ANONYMOUS_USER_NAME`).

Você pode remover funções deste grupo para limitar as permissões para usuários não autenticados.

Funções padrão: *Adicionar sugestão, Acessar repositório*

Visualizadores Essa função garante a visibilidade de projetos públicos para todos os usuários. Por padrão, todos os usuários 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 padrão: nenhuma

Usuários Grupo padrão para todos os usuários.

Por padrão, *associação automática de grupo* faz com que todos os novos membros de contas deste grupo quando eles entram.

Funções padrão: *Usuário avançado*

Revisores Grupo para revisores (consulte *Fluxos de trabalho de tradução*).

Funções padrão: *Revisar textos*

Gerenciadores Grupo pra administradores.

Funções padrão: *Administração*

Aviso: Nunca remova os grupos e usuários predefinidos do Weblate, pois isso pode levar a problemas inesperados! Se você não tiver uso para eles, você pode remover todos os privilégios deles.

2.6.3 Restrições adicionais de acesso

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

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

2.7 Projetos de tradução

2.7.1 Translation organization

Weblate organizes translatable VCS content of project/components into a tree-like structure.

- The bottom level object is *Configuração de projeto*, which should hold all translations belonging together (for example translation of an application in several versions and/or accompanying documentation).
- On the level above, *Configuração de componente*, which is actually the component to translate, you define the VCS repository to use, and the mask of files to translate.
- Above *Configuração de componente* there are individual translations, handled automatically by Weblate as translation files (which match *Máscara de arquivo* defined in *Configuração de componente*) appear in the VCS repository.

Weblate supports a wide range of translation formats (both bilingual and monolingual ones) supported by Translate Toolkit, see *Formatos de arquivos suportados*.

Nota: You can share cloned VCS repositories using *URLs internas do Weblate*. Using this feature is highly recommended when you have many components sharing the same VCS. It improves performance and decreases required disk space.

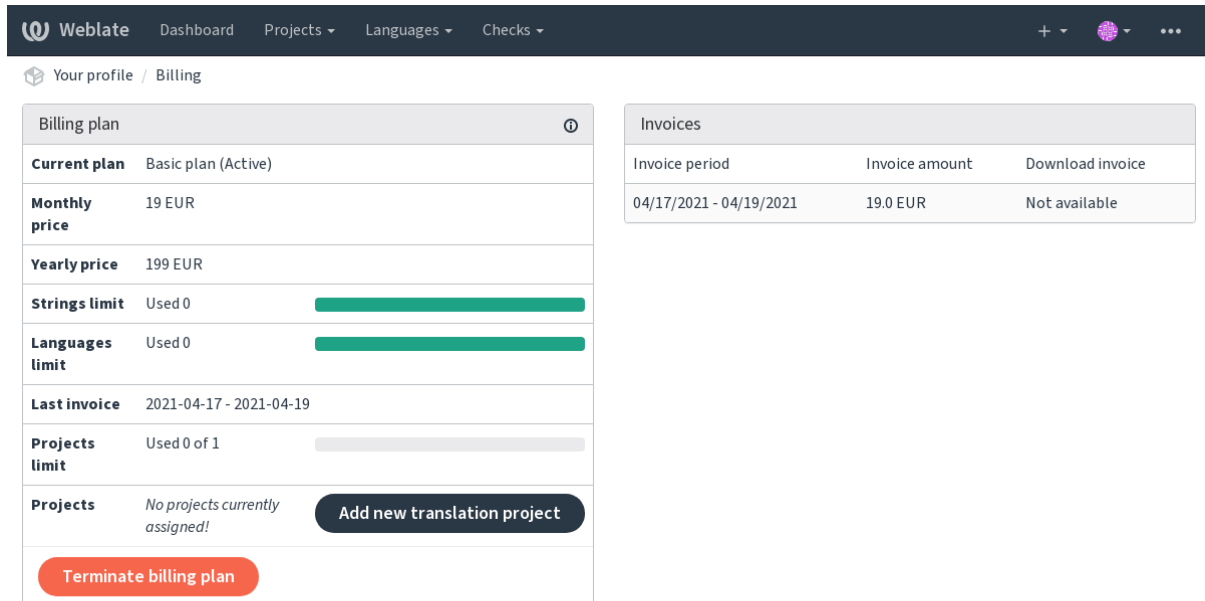
2.7.2 Adicionando projetos e componentes de tradução

Alterado na versão 3.2: An interface for adding projects and components is included, and you no longer have to use *A interface administrativa do Django*.

Alterado na versão 3.4: The process of adding components is now multi staged, with automated discovery of most parameters.

Based on your permissions, new translation projects and components can be created. It is always permitted for users with the *Add new projects* permission, and if your instance uses billing (e.g. like <https://hosted.weblate.org/> see *Cobrança*), you can also create those based on your plans allowance from the user account that manages billing.

You can view your current billing plan on a separate page:

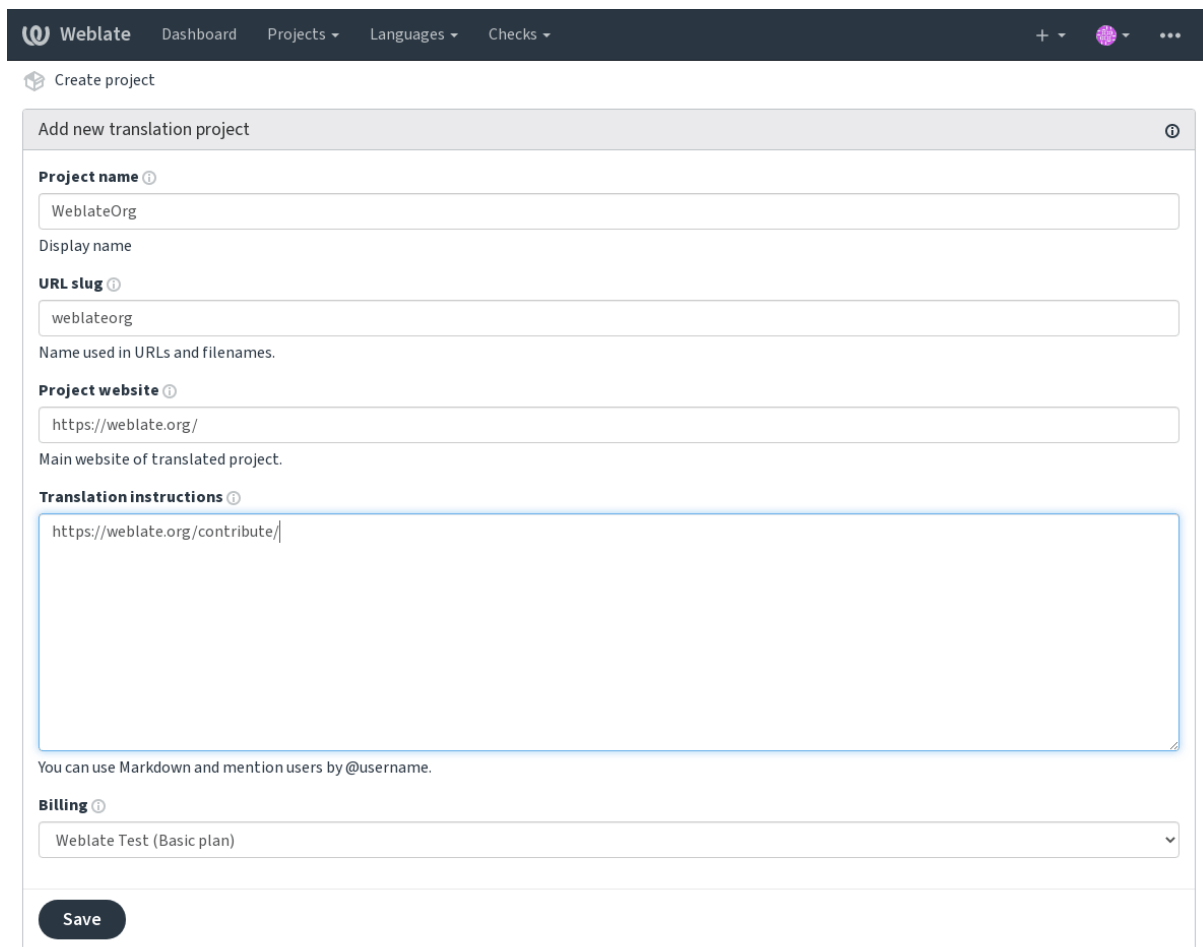


Billing plan	
Current plan	Basic plan (Active)
Monthly price	19 EUR
Yearly price	199 EUR
Strings limit	Used 0 <div></div>
Languages limit	Used 0 <div></div>
Last invoice	2021-04-17 - 2021-04-19
Projects limit	Used 0 of 1 <div></div>
Projects	No projects currently assigned! <button>Add new translation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

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

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

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



The screenshot shows the 'Add new translation project' form in the Weblate web interface. The form is titled 'Add new translation project' and includes several input fields and a text area. The 'Project name' field contains 'WeblateOrg'. The 'URL slug' field contains 'weblateorg'. The 'Project website' field contains 'https://weblate.org/'. The 'Translation instructions' text area contains 'https://weblate.org/contribute/'. The 'Billing' dropdown menu is set to 'Weblate Test (Basic plan)'. A 'Save' button is at the bottom of the form. The form also includes a 'Create project' button at the top left and a 'Powered by Weblate 4.6' footer at the bottom.

Create project

Add new translation project

Project name ⓘ

WeblateOrg

Display name

URL slug ⓘ

weblateorg

Name used in URLs and filenames.

Project website ⓘ

https://weblate.org/

Main website of translated project.

Translation instructions ⓘ

https://weblate.org/contribute/

You can use Markdown and mention users by @username.

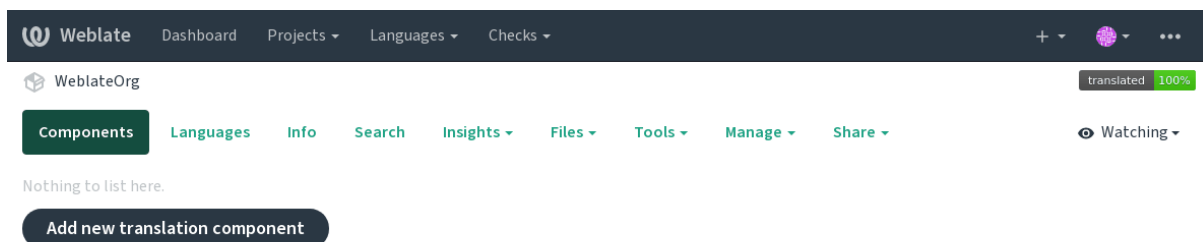
Billing ⓘ

Weblate Test (Basic plan)

Save

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

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



The screenshot shows the project page for 'WeblateOrg' in the Weblate web interface. The page has a dark header with the Weblate logo and navigation links: 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below the header, there is a 'WeblateOrg' section with a 'translated 100%' badge. A 'Components' button is highlighted in green. Below this, there is a 'Nothing to list here.' message and an 'Add new translation component' button. The footer of the page includes 'Powered by Weblate 4.6' and several links: 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Weblate

Dashboard Projects Languages Checks

WeblateOrg

translated 100%

Components Languages Info Search Insights Files Tools Manage Share

Watching

Nothing to list here.

Add new translation component

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

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

De controle de versão Creates component from remote version control repository.

De componente existente Creates additional component to existing one by choosing different files.

Ramo adicional Creates additional component to existing one, just for different branch.

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

Traduzir documento Upload single document and translate that.

Iniciar do zero Create blank translation project and add strings manually.

Once you have existing translation components, you can also easily add new ones for additional files or branches using same repository.


First you need to fill in name and repository location:

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

On the next page, you are presented with a list of discovered translatable resources:

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

As a last step, you review the translation component info and fill in optional details:

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

Ver também:

A interface administrativa do Django, Configuração de projeto, Configuração de componente

2.7.3 Configuração de projeto

Create a translation project and then add a new component for translation in it. The project is like a shelf, in which real translations are stacked. All components in the same project share suggestions and their dictionary; the translations are also automatically propagated through all components in a single project (unless turned off in the component configuration), see *Memória de tradução*.

Ver também:

/devel/integration

These basic attributes set up and inform translators of a project:

Nome do projeto

Verbose project name, used to display the project name.

URL amigável

Project name suitable for URLs.

Site do projeto

URL where translators can find more info about the project.

This is a required parameter unless turned off by *WEBSITE_REQUIRED*.

Instruções de tradução

URL to more site with more detailed instructions for translators.

Definir o cabeçalho do “Language-Team”

Whether Weblate should manage the Language-Team header (this is a *GNU gettext* only feature right now).

Usar memória de tradução compartilhada

Whether to use shared translation memory, see *Memória de tradução compartilhada* for more details.

Default value is determined by *DEFAULT_SHARED_TM*.

Contribuir com memória de tradução compartilhada

Whether to contribute to shared translation memory, see *Memória de tradução compartilhada* for more details.

Default value is determined by `DEFAULT_SHARED_TM`.

Controle de acesso

Configure per project access control, see *Controle de acesso do projeto* for more details.

Default value can be changed by `DEFAULT_ACCESS_CONTROL`.

Habilitar revisões

Enable review workflow for translations, see *Revisores dedicados*.

Habilitar revisões de fontes

Enable review workflow for source strings, see *Revisões de textos fonte*.

Ver também:

report-source, *Comentários*

Habilitar ganchos

Whether unauthenticated *Ganchos de notificação* are to be used for this repository.

Ver também:

Arquivo de idioma intermediário, *Rota de qualidade para os textos fonte*, *Formatos bilíngues e monolíngues*, *Definições de idioma*

Aliases de idioma

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

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

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

Using non standard code: `ia_FOO:ia`

Dica: Os códigos de idioma são mapeados ao combinar os arquivos 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 arquivos.

Ver também:

Analisando códigos de idioma

2.7.4 Configuração de componente

A component is a grouping of something for translation. You enter a VCS repository location and file mask for which files you want translated, and Weblate automatically fetches from this VCS, and finds all matching translatable files.

Ver também:

/devel/integration

You can find some examples of typical configurations in the *Formatos de arquivos suportados*.

Nota: It is recommended to keep translation components to a reasonable size - split the translation by anything that makes sense in your case (individual apps or addons, book chapters or websites).

Weblate easily handles translations with 10000s of strings, but it is harder to split work and coordinate among translators with such large translation components.

Should the language definition for a translation be missing, an empty definition is created and named as “cs_CZ (generated)”. You should adjust the definition and report this back to the Weblate authors, so that the missing languages can be included in next release.

The component contains all important parameters for working with the VCS, and for getting translations out of it:

Nome do componente

Verbose component name, used to display the component name.

URL amigável do componente

Component name suitable for URLs.

Component project

Configuração de projeto where the component belongs.

Sistema de controle de versão

VCS to use, see *Integração com controle de versão* for details.

Ver também:

Fazendo push das alterações do Weblate

Repositório do código-fonte

VCS repository used to pull changes.

Ver também:

See *Acessando repositórios* for more details on specifying URLs.

Dica: This can either be a real VCS URL or `weblate://project/component` indicating that the repository should be shared with another component. See *URLs internas do Weblate* for more details.

URL de push do repositório

Repository URL used for pushing. This setting is used only for *Git* and *Mercurial* and push support is turned off for these when this is empty.

Ver também:

See *Acessando repositórios* for more details on how to specify a repository URL and *Fazendo push das alterações do Weblate* for more details on pushing changes from Weblate.

Navegador do repositório

URL of repository browser used to display source files (location of used messages). When empty, no such links will be generated. You can use *Template markup*.

For example on GitHub, use something like: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

In case your paths are relative to different folder, you might want to strip leading directory by parent-dir filter (see *Template markup*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

URL do repositório exportado

URL where changes made by Weblate are exported. This is important when *Localização contínua* is not used, or when there is a need to manually merge changes. You can use *Exportador git* to automate this for Git repositories.

Ramo do repositório

Which branch to checkout from the VCS, and where to look for translations.

Ramo do push

Branch for pushing changes, leave empty to use *Ramo do repositório*.

Nota: This is currently only supported for Git, GitLab and GitHub, it is ignored for other VCS integrations.

Ver também:

Fazendo push das alterações do Weblate

Máscara de arquivo

Mask of files to translate, including path. It should include one “*” replacing language code (see *Definições de idioma* for info on how this is processed). In case your repository contains more than one translation file (e.g. more gettext domains), you need to create a component for each of them.

For example `po/* .po` or `locale/*/LC_MESSAGES/django.po`.

In case your filename contains special characters such as `[]`, these need to be escaped as `[]` or `[]`.

Ver também:

Formatos bilíngues e monolíngues, O que significa “Existem mais arquivos para o idioma único (en)”?

Arquivo de idioma da base monolíngue

Base file containing string definitions for *Componentes monolíngues*.

Ver também:

Formatos bilíngues e monolíngues, O que significa “Existem mais arquivos para o idioma único (en)”?

Editar o arquivo base

Whether to allow editing the base file for *Componentes monolíngues*.

Arquivo de idioma intermediário

Intermediate language file for *Componentes monolíngues*. In most cases this is a translation file provided by developers and is used when creating actual source strings.

When set, the source strings are based on this file, but all other languages are based on *Arquivo de idioma da base monolíngue*. In case the string is not translated into the source language, translating to other languages is prohibited. This provides *Rota de qualidade para os textos fonte*.

Ver também:

Rota de qualidade para os textos fonte, Formatos bilíngues e monolíngues, O que significa “Existem mais arquivos para o idioma único (en)”?

Modelo para novas traduções

Base file used to generate new translations, e.g. `.pot` file with `gettext`.

Dica: In many monolingual formats Weblate starts with blank file by default. Use this in case you want to have all strings present with empty value when creating new translation.

Ver também:

adding-translation, Adicionando novas traduções, Adicionando nova tradução, Formatos bilíngues e monolíngues, O que significa “Existem mais arquivos para o idioma único (en)”?

Formato de arquivo

Translation file format, see also *Formatos de arquivos suportados*.

Endereço do relatório de erros do texto fonte

Email address used for reporting upstream bugs. This address will also receive notification about any source string comments made in Weblate.

Permitir propagação de tradução

You can turn off propagation of translations to this component from other components within same project. This really depends on what you are translating, sometimes it's desirable to have make use of a translation more than once.

It's usually a good idea to turn this off for monolingual translations, unless you are using the same IDs across the whole project.

Default value can be changed by `DEFAULT_TRANSLATION_PROPAGATION`.

Habilitar sugestões

Whether translation suggestions are accepted for this component.

Votação de sugestões

Turns on vote casting for suggestions, see *Votação de sugestões*.

Aceitar sugestões automaticamente

Automatically accept voted suggestions, see *Votação de sugestões*.

Marcadores de tradução

Customization of quality checks and other Weblate behavior, see *Personalizando comportamento usando marcadores*.

Verificações forçadas

List of checks which can not be ignored, see *Forçando verificações*.

Nota: Enforcing the check does not automatically enable it, you still should enabled it using *Personalizando comportamento usando marcadores* in *Marcadores de tradução* or *Informações adicionais sobre textos fonte*.

Licença da tradução

License of the translation (does not need to be the same as the source code license).

Acordo de colaborador

Acordo de usuário que precisa ser aprovado antes de um usuário poder traduzir este componente.

Adicionando nova tradução

How to handle requests for creation of new languages. Available options:

Contatar os mantenedores User can select desired language and the project maintainers will receive a notification about this. It is up to them to add (or not) the language to the repository.

Apontar para a URL de instruções de tradução User is presented a link to page which describes process of starting new translations. Use this in case more formal process is desired (for example forming a team of people before starting actual translation).

Criar novo arquivo de idioma User can select language and Weblate automatically creates the file for it and translation can begin.

Desabilitar adição de novas traduções There will be no option for user to start new translation.

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).

Ver também:

adding-translation, *Adicionando novas traduções*

Gerenciar textos

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, os textos são gerenciados apenas no idioma fonte e são automaticamente adicionados ou removidos nas traduções. Os textos aparecem nos arquivos de tradução assim que são traduzidos.

Ver também:

Formatos bilíngues e monolíngues, adding-new-strings, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Estilo de código de idioma

Personalize o código de idioma usado para gerar o nome do arquivo para traduções criado por Weblate.

Ver também:

Adicionando novas traduções, *Código do idioma*, *Analizando códigos de idioma*

Estilo de mesclagem

You can configure how updates from the upstream repository are handled. This might not be supported for some VCSs. See *Merge 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`.

Push ao fazer commit

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 para 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 em 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 [Arquivo de idioma intermediário](#).

Filtro de idioma

Regular expression used to filter the translation when scanning for filemask. It can be used to limit the list of languages managed by Weblate.

Nota: You need to list language codes as they appear in the filename.

Some examples of filtering:

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 de variantes

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

Nota: Most of the fields can be edited by project owners or managers, in the Weblate interface.

Ver também:

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

Prioridade

Componentes com prioridade mais alta são oferecidos primeiro para os 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.

Allows using this component as a glossary. You can configure how it will be listed using *Cor do glossário*.

The glossary will be accessible in all projects defined by *Compartilhar em projetos*.

It is recommended to enable *Gerenciar textos* on glossaries in order to allow adding new words to them.

Ver 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 componente*
 - *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 arquivo 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 {{file-
name|parentdir}}
```

You can combine them with filters:

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


You can use conditions:

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

There is additional tag available for replacing characters:

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

You can combine it with filters:

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

There are also additional filter to manipulate with filenames:

```
Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|striptext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times: {{ filename|parentdir|parentdir }}
```

...and other Django template features.

2.7.6 Importing speed

Fetching VCS repository and importing translations to Weblate can be a lengthy process, depending on size of your translations. Here are some tips:

Optimize configuration

The default configuration is useful for testing and debugging Weblate, while for a production setup, you should do some adjustments. Many of them have quite a big impact on performance. Please check [Configuração de produção](#) for more details, especially:

- Configure Celery for executing background tasks (see [Tarefas de fundo usando Celery](#))
- *Habilitar o cache*
- *Usar um poderoso mecanismo de banco de dados*
- *Desabilitar 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 usando 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 componente* addon.

To use the addon, you first need to create a component for one translation file (choose the one that is the least likely to be renamed or removed in future), and install the addon on this component.

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

Ver também:

Comandos de gerência, Descoberta de componente

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.

Você pode ajustar ainda mais esse mapeamento no nível do projeto por *Aliases de idioma*.

Se nenhuma correspondência exata for encontrada, uma tentativa será feita para melhor encaixá-la em um idioma existente. São testadas etapas a seguir:

- Procuras sem diferenciar maiúsculo de minúsculo.
- Normalizar sublinhados e traços.
- Looking up built-in language aliases.
- Procurar por nome de idioma.
- Ignorar o código de país padrão para o idioma dado – escolher `cs` em vez de `cs_CZ`.

Should that also fail, a new language definition will be created using the defaults (left to right text direction, one plural). The automatically created language with code `xx_XX` will be named as `xx_XX (generated)`. You might want to change this in the admin interface later, (see *Changing language definitions*) and report it to the issue tracker (see *Contribuindo para o Weblate*), so that the proper definition can be added to the upcoming Weblate release.

Dica: In case you see something unwanted as a language, you might want to adjust *Filtro de idioma* to ignore such file when parsing translations.

Ver também:

Código do idioma, Adicionando novas traduções

2.8.2 Changing language definitions

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

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

2.8.3 Built-in language definitions

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

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

Ver também:

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

2.8.4 Ambiguous language codes and macrolanguages

In many cases it is not a good idea to use macro language code for a translation. The typical problematic case might be Kurdish language, which might be written in Arabic or Latin script, depending on actual variant. To get correct behavior in Weblate, it is recommended to use individual language codes only and avoid macro languages.

Ver também:

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

2.8.5 Definições de idioma

Each language consists of following fields:

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](#).

Ver também:

Analisando códigos de idioma, *Adicionando novas traduções*

Nome do idioma

Visible name of the language. The language names included in Weblate are also being localized depending on user interface language.

Direção do texto

Determines whether language is written right to left or left to right. This property is autodetected correctly for most of the languages.

Plural number

Number of plurals used in the language.

Fórmula de plural

Gettext compatible plural formula used to determine which plural form is used for given count.

Ver 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: In versions prior to 2.18 the behaviour of adding new translations was file format specific.

Weblate can automatically start new translation for all of the file formats.

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

When you specify *Modelo para novas traduções* in *Configuração de componente*, Weblate will use this file to start new translations. Any exiting translations will be removed from the file when doing so.

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

The *Estilo de código de idioma* allows you to customize language code used in generated filenames:

Padrão baseado no formato de arquivo Dependent on file format, for most of them POSIX is used.

Estilo POSIX usando sublinhado como um separador Typically used by gettext and related tools, produces language codes like `pt_BR`.

Estilo POSIX usando sublinhado como separador, incluindo código de país POSIX style language code including the country code even when not necessary (for example `cs_CZ`).

Estilo BCP usando hífen como um separador Typically used on web platforms, produces language codes like `pt-BR`.

Estilo BCP usando hífen como separador, incluindo código de país BCP style language code including the country code even when not necessary (for example `cs-CZ`).

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

Estilo Java Used by Java—mostly BCP with legacy codes for Chinese.

Additionally, any mappings defined in *Aliases de idioma* are applied in reverse.

Nota: Weblate recognizes any of these when parsing translation files, the above settings only influences how new files are created.

Ver também:

Código do idioma, *Analisando códigos de idioma*

2.9 Localização contínua

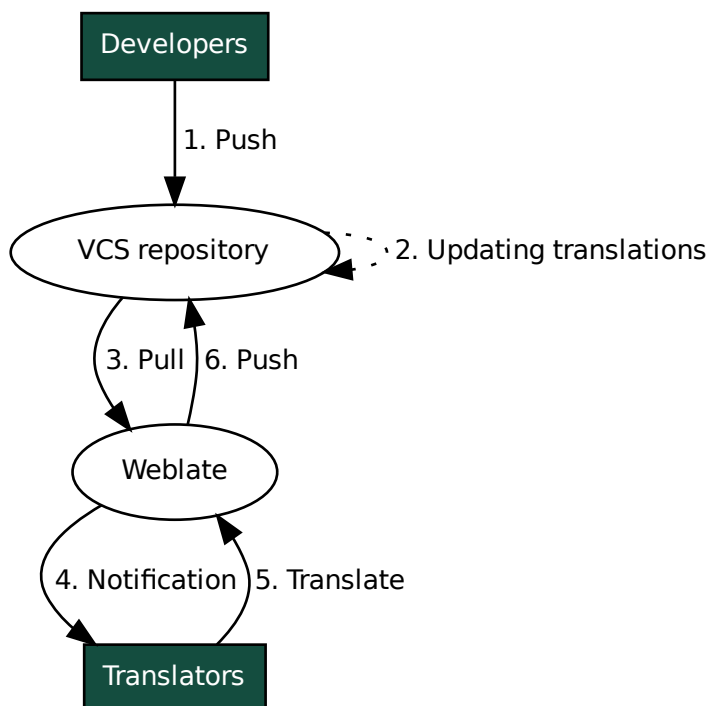
Há infraestrutura em vigor para que sua tradução acompanhe de perto o desenvolvimento. Desta forma, os tradutores podem trabalhar em traduções o tempo todo, em vez de trabalhar através de uma enorme quantidade de texto novo pouco antes do lançamento.

Ver também:

`/devell/integration` descreve maneiras básicas de integrar seu desenvolvimento com o Weblate.

Este é o processo:

1. Os desenvolvedores fazem alterações e fazem *push* delas para o repositório VCS.
2. Opcionalmente, os arquivos de tradução são atualizados (isso depende do formato do arquivo, consulte *Por que é que o Weblate ainda mostra os textos de tradução antigos quando eu atualizei o modelo?*).
3. O Weblate faz o *pull* das alterações do repositório VCS, consulte *Atualizando repositórios*.
4. Uma vez que o Weblate detecta alterações nas traduções, os tradutores são notificados com base em suas configurações de assinatura.
5. Os tradutores enviam traduções usando a interface web do Weblate ou enviam alterações feitas offline.
6. Uma vez que os tradutores tenham finalizado, 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 Atualizando repositórios

Você deve configurar alguma maneira de atualizar repositórios de backend a partir de sua fonte.

- Use *Ganchos de notificação* para integrar com a maioria dos serviços comuns de hospedagem de código:
 - *Recebendo automaticamente alterações do GitHub*
 - *Recebendo automaticamente alterações do GitLab*
 - *Recebendo automaticamente alterações do Bitbucket*
 - *Recebendo automaticamente alterações do Pagure*
 - *Recebendo automaticamente alterações do Azure Repos*
- Acione manualmente a atualização no gerenciamento do repositório ou usando *API REST do Weblate* ou *Weblate Client*
- Habilite *AUTO_UPDATE* para atualizar automaticamente todos os componentes na sua instância Weblate
- Execute *updategit* (com a seleção de um projeto ou *-all* para atualizar tudo)

Sempre que o Weblate atualizar o repositório, as extensões de pós-atualização serão acionadas, consulte *Extensões*.

Evitando conflitos de mesclagem

Os conflitos de mesclagem do Weblate surgem quando o mesmo arquivo 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 ele descarte alterações antes de atualizar os arquivos fora do Weblate.

A primeira abordagem é fácil com arquivos monolíngues - você pode adicionar novos textos no Weblate e deixar a edição completa dos arquivos lá. Para arquivos bilíngues, geralmente há algum tipo de processo de extração de mensagens para gerar arquivos traduzíveis a partir 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 usando **xgettext**) e, em seguida, o processo mais mescla-o em traduções reais (os arquivos GETTEXT PO são atualizados usando **msgmerge**). Você pode executar o segundo passo dentro do Weblate e ele garantirá que todas as alterações pendentes sejam incluídas antes desta operação.

A segunda abordagem pode ser alcançada usando *API REST do Weblate* para forçar o Weblate a fazer push de todas as alterações pendentes e bloquear a tradução enquanto você 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 você tiver vários componentes compartilhando o mesmo repositório, você precisa bloqueá-los todos separadamente:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Nota: O exemplo usa *Weblate Client*, que precisa de configuração (chaves de API) para ser capaz de controlar o Weblate remotamente. Você também pode conseguir isso usando qualquer cliente HTTP em vez de wlc, por exemplo, curl, ver *API REST do Weblate*.

Ver também:

Weblate Client

Recebendo automaticamente alterações do GitHub

O Weblate vem com suporte nativo ao GitHub.

Se você estiver usando o Hosted Weblate, a abordagem recomendada é instalar o [aplicativo Weblate](#), dessa forma você terá a configuração correta sem ter que configurar muita coisa. 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 repository settings page for 'WeblateOrg / hello'. The 'Webhooks' section is active, and the 'Add webhook' form is displayed. The 'Payload URL' is set to 'https://hosted.weblate.org/hooks/github/'. The 'Content type' is 'application/x-www-form-urlencoded'. The 'Secret' field is empty. The 'Which events would you like to trigger this webhook?' section has 'Just the push event.' selected. The 'Active' checkbox is checked. The 'Add webhook' button is green.

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/`.

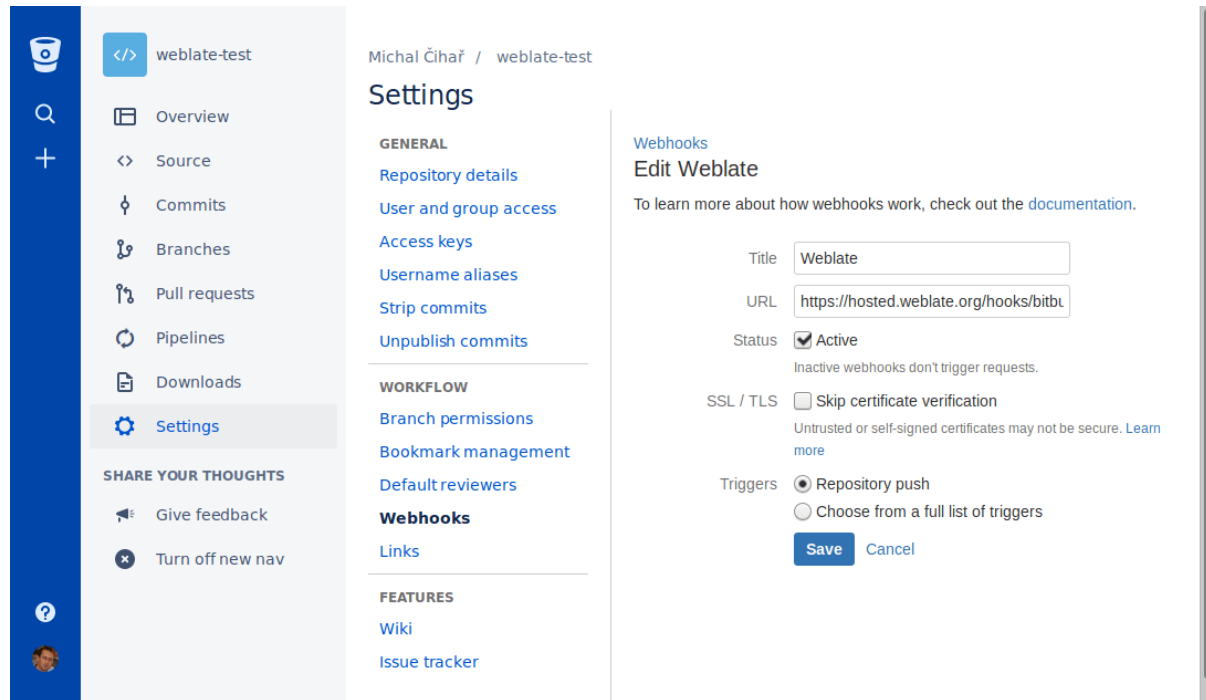
Você pode deixar outros valores nas configurações padrão (o Weblate pode lidar com ambos os tipos de conteúdo e consome apenas o evento *push*).

Ver também:

POST /hooks/github/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do Bitbucket

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/`).

**Ver também:**

POST /hooks/bitbucket/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do GitLab

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/`).

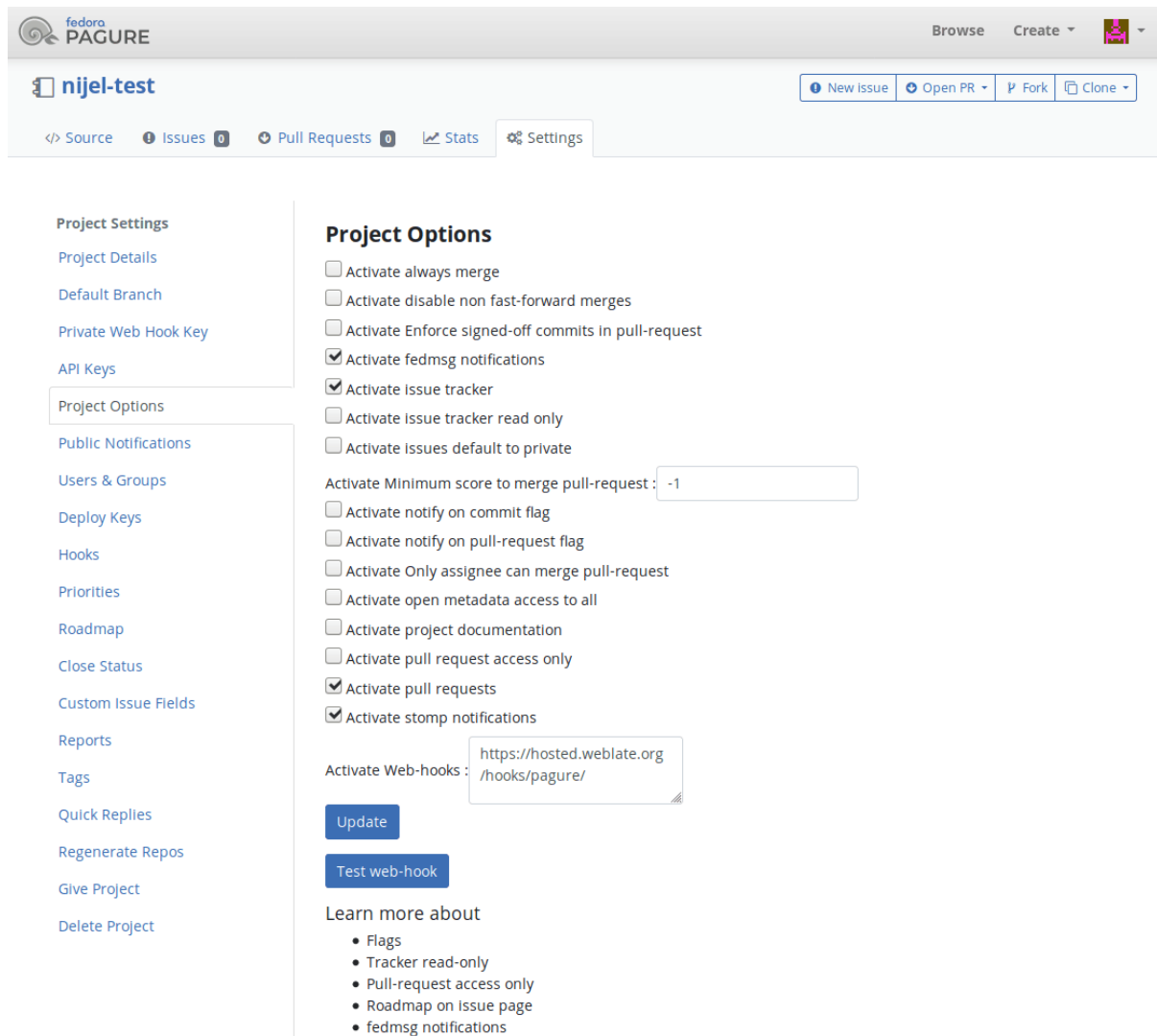
Ver também:

POST /hooks/gitlab/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do Pagure

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 the navigation bar, there are tabs for 'Source', 'Issues', 'Pull Requests', 'Stats', and 'Settings'. The 'Settings' tab is active, showing a sidebar with various settings categories like 'Project Settings', 'Project Details', 'Default Branch', 'Private Web Hook Key', 'API Keys', 'Project Options', 'Public Notifications', 'Users & Groups', 'Deploy Keys', 'Hooks', 'Priorities', 'Roadmap', 'Close Status', 'Custom Issue Fields', 'Reports', 'Tags', 'Quick Replies', 'Regenerate Repos', 'Give Project', and 'Delete Project'. The 'Project Options' section is expanded, displaying a list of checkboxes for various features: 'Activate always merge', 'Activate disable non fast-forward merges', 'Activate Enforce signed-off commits in pull-request', 'Activate fedmsg notifications' (checked), 'Activate Issue tracker' (checked), 'Activate Issue tracker read only', 'Activate Issues default to private', 'Activate Minimum score to merge pull-request' (set to -1), 'Activate notify on commit flag', 'Activate notify on pull-request flag', 'Activate Only assignee can merge pull-request', 'Activate open metadata access to all', 'Activate project documentation', 'Activate pull request access only', 'Activate pull requests' (checked), and 'Activate stomp notifications' (checked). Below these options, there is a section for 'Activate Web-hooks' with a text input field containing 'https://hosted.weblate.org/hooks/pagure/' and buttons for 'Update' and 'Test web-hook'. At the bottom, there is a 'Learn more about' section with links to 'Flags', 'Tracker read-only', 'Pull-request access only', 'Roadmap on Issue page', and 'fedmsg notifications'.

Ver também:

POST /hooks/pagure/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do Azure Repos

Novo na versão 3.8.

O Weblate tem suporte para webhooks do Azure Repos, adicione um webhook para 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*.

Ver também:

Webhooks no manual do Azure DevOps, *POST /hooks/azure/, Acessando repositórios do Hosted Weblate*

Recebendo automaticamente alterações do Gitea Repos

Novo na versão 3.9.

Weblate tem suporte para webhooks do Gitea, adicione um *Gitea Webhook* para evento *Push events* com destino para a 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.

Ver também:

Webhooks no manual do Gitea, *POST /hooks/gitea/*, *Acessando repositórios do Hosted Weblate*

Recebendo automaticamente alterações de Gitee Repos

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.

Ver também:

Webhooks no manual do Gitee, *POST /hooks/gitee/*, *Acessando repositórios do Hosted Weblate*

Atualizando automaticamente repositórios *nightly*

O Weblate busca automaticamente repositórios remotos *nightly* para melhorar o desempenho ao mesclar alterações mais tarde. Você 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 push do repositório*) e, nesse caso, o Weblate será capaz de fazer push de alteração para o repositório remoto. O Weblate também pode ser configurado para fazer push automaticamente das alterações em cada commit (isso é o padrão, veja *Push ao fazer commit*). Se você não quiser que seja feito push automático das alterações, você pode fazer isso manualmente em *Manutenção do repositório* ou usando API via *wlc push*.

As opções de push diferem com base no *Integração com controle de versão* usado, mais detalhes são encontrados nesse capítulo.

No caso de você não querer pushes diretos pelo Weblate, há suporte para pull requests do *GitHub*, do *GitLab* e do *Pagure* ou revisões do *Gerrit*, você pode ativá-los escolhendo *GitHub*, *GitLab*, *Gerrit* ou *Pagure*, como *Sistema de controle de versão* em *Configuração de componente*.

No geral, as opções a seguir estão disponíveis com Git, GitHub e GitLab:

Configuração desejada	Sistema de controle de versão	URL de push do repositório	Ramo do push
Sem push	<i>Git</i>	<i>vazio</i>	<i>vazio</i>
Push diretamente	<i>Git</i>	URL de SSH	<i>vazio</i>
Fazer push para ramo separado	<i>Git</i>	URL de SSH	Nome do ramo
Pull request de GitHub do fork	<i>GitHub</i>	<i>vazio</i>	<i>vazio</i>
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>	<i>vazio</i>	<i>vazio</i>
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>	<i>vazio</i>	<i>vazio</i>
Merge request de Pagure do ramo	<i>Pagure</i>	URL de SSH ^{Página 62, 1}	Nome do ramo

Nota: Você também pode habilitar o push automático de alterações após o Weblate fazer commit, isso pode ser feito em *Push ao fazer commit*.

Ver também:

Consulte *Acessando repositórios* para configurar chaves SSH e *Commits adiados* para obter informações sobre quando o Weblate decide fazer commit de alterações.

Ramos protegidos

Se você estiver usando o Weblate em ramo protegido, você pode configurá-lo para usar pull requests e executar revisão real sobre as traduções (o que pode ser problemático para idiomas que você não conhece). Uma abordagem alternativa é abrir mão desta limitação em favor do usuário 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 Merge ou rebase

Por padrão, o Weblate mescla o repositório upstream em seu próprio. Esta é a maneira mais segura no caso de você também acessar o repositório subjacente por outros meios. Caso você não precise disso, você 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 você quer ou não habilitá-los.

2.9.4 Interagindo com os outros

O Weblate facilita a interação com outras pessoas usando sua API.

Ver também:

API REST do Weblate

2.9.5 Commits adiados

O comportamento do Weblate é agrupar commits do mesmo autor em um só commit, se possível. Isso reduz consideravelmente o número de commits, no entanto, você pode precisar dizer explicitamente para ele fazer os commits no caso de você querer deixar o repositório VCS em sincronia, por exemplo, para mesclagem (isso é por padrão permitido para o grupo *Gerenciadores*, consulte [Lista de privilégios](#)).

As alterações neste modo têm seu commit feito assim que qualquer uma das seguintes condições são cumpridas:

- Outra pessoa altera um texto já alterado.
- 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 para fazer commit* em *Configuração de componente*.

Dica: Os commits são criados para cada componente. Então, caso você tenha muitos componentes, você ainda verá muitos compromissos. Você pode utilizar a extensão *Squash de commits git* neste caso.

Se você quiser fazer commit das alterações com mais frequência e sem verificar a idade, você 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 Processando 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 Mantendo traduções iguais entre componentes

Uma vez que você tenha vários componentes de tradução, você pode querer garantir que os mesmos textos tenham a mesma tradução. Isso pode ser alcançado em vários níveis.

Propagação de tradução

Com a propagação de tradução habilitada (que é o padrão, consulte [Configuração de componente](#)), todas as novas traduções são feitas automaticamente em todos os componentes com textos correspondentes. Estas traduções são devidamente creditadas ao usuário 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 os textos são diferentes. Você 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. Você pode acioná-la manualmente (veja [Tradução automática](#)) ou fazê-la ser executada automaticamente na atualização do repositório usando uma extensão (veja [Tradução automática](#)).

2.10 Licenciando traduções

Você pode especificar sob quais traduções de licença são contribuídas. Isso é especialmente importante de se as traduções forem abertas ao público, para estipular para que elas possam ser usadas.

Você deve especificar as informações da licença da [Configuração de componente](#). Você 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 seção de informações de tradução da respectiva [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, você provavelmente precisa de consentimento prévio.

2.10.2 Acordo de colaborador

Se você especificar um contrato de licença de colaborador, apenas os usuários que concordaram com ele poderão contribuir. Este é um passo claramente visível ao acessar a tradução:

Contribution to this translation requires you to agree with a contributor agreement. [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 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

O texto inserido é formatado em parágrafos e links externos podem ser incluídos. A marcação HTML não pode ser usada.

2.10.3 Licenças de usuário

Qualquer usuário pode rever todas as licenças de tradução de todos os projetos públicos na instância a partir de seu perfil:

The screenshot shows the Weblate user interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks' menus. On the right are icons for a wrench, a plus sign, a user profile, and a menu. Below this is a light blue bar with 'Your profile' and a list of tabs: 'Languages', 'Preferences', 'Notifications', 'Account', 'Profile', 'Licenses' (which is highlighted in dark blue), 'Audit log', and 'API access'. The main content area has a header 'Licenses' and a text block explaining licensing: '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:'. A bulleted list shows 'WeblateOrg/Language names'. Below this is a section 'Licenses for individual translations' containing 'GNU General Public License v3.0 or later' with a 'GPL-3.0' badge and links to 'WeblateOrg/WebateOrg', 'WeblateOrg/Djangojs', 'WeblateOrg/Django', and 'WeblateOrg/Language names'. It also lists 'MIT License' with a 'MIT' badge and a link to 'WeblateOrg/Android'. At the bottom of the page is a footer: 'Powered by Weblate 4.6 About Weblate Legal Contact Documentation Donate to Weblate'.

2.11 Processo de tradução

2.11.1 Votação de sugestões

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* configuration with *Suggestion voting* to turn on voting, and *Autoaccept suggestions* to set a threshold for accepted suggestions (this includes a vote from the user making the suggestion if it is cast).

Nota: Once automatic acceptance is set up, normal users lose the privilege to directly save translations or accept suggestions. This can be overridden with the *Edit string when suggestions are enforced permission*.

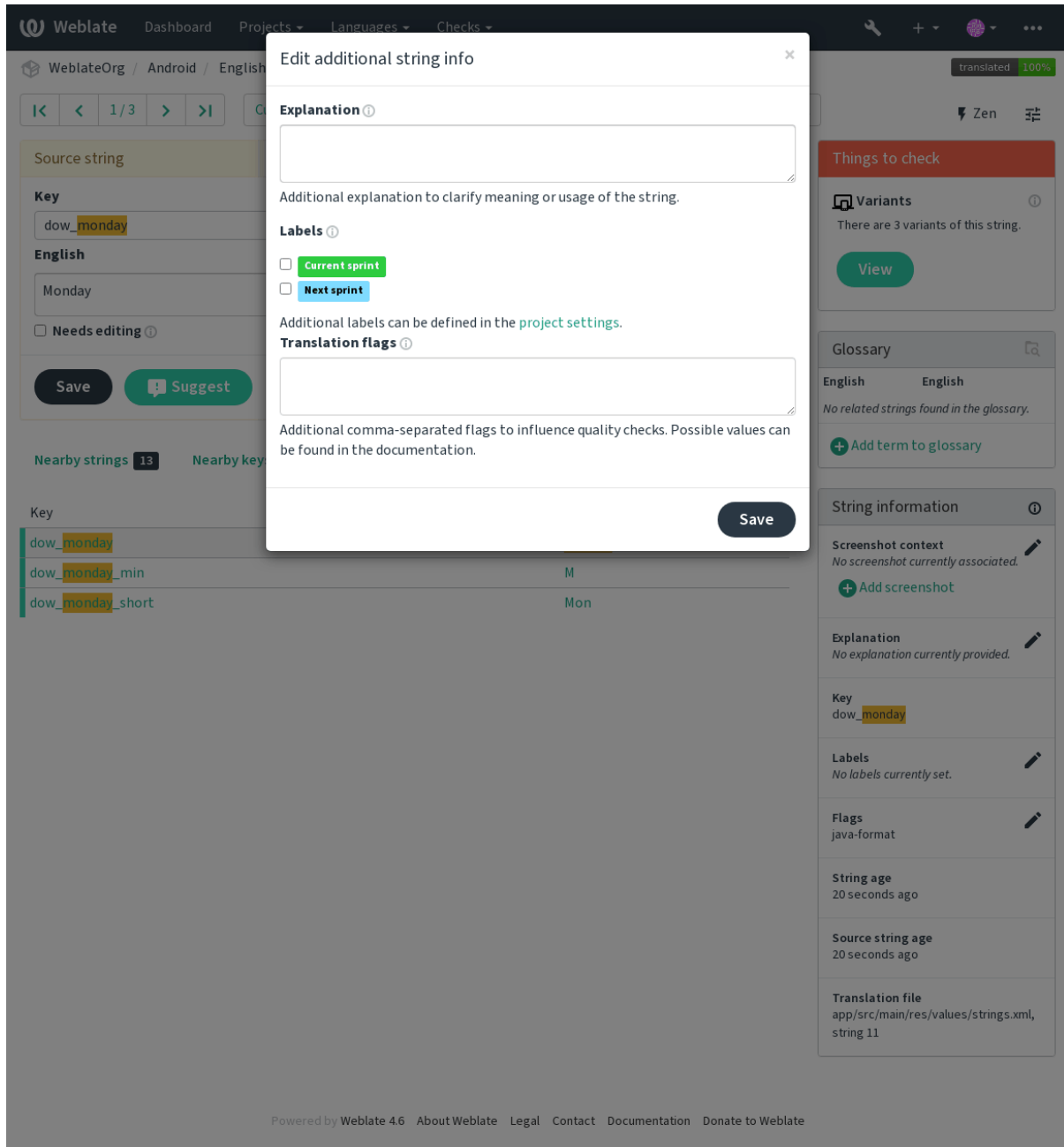
You can combine these with *access control* into one of the following setups:

- Users suggest and vote for suggestions and a limited group controls what is accepted. - Turn on voting. - Turn off automatic acceptance. - Don't let users save translations.
- Users suggest and vote for suggestions with automatic acceptance once the defined number of them agree. - Turn on voting. - Set the desired number of votes for automatic acceptance.

- Optional voting for suggestions. (Can optionally be used by users when they are unsure about a translation by making multiple suggestions.) - Only turn on voting.

2.11.2 Informações adicionais sobre textos 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*.

Webplate

Dashboard

Projects

Languages

Checks

WebplateOrg

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

weblate.org

translation

překlad

weblate.org

project

projekt

weblate.org

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

Help text for automatic translation tool

Labels

No labels currently set.

Flags

No flags currently set.

Source string location

weblate/templates/translation.html:212

String age

3 seconds ago

Source string age

3 seconds ago

Translation file

weblate/locale/cs/LC_MESSAGES/django.po, string 11

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

262

Capítulo 2. Documentação de administrador

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.

Ver 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.

Ver também:

Verificações de qualidade

Explicação

Alterado na versão 4.1: In previous versions this has been called *Extra context*.

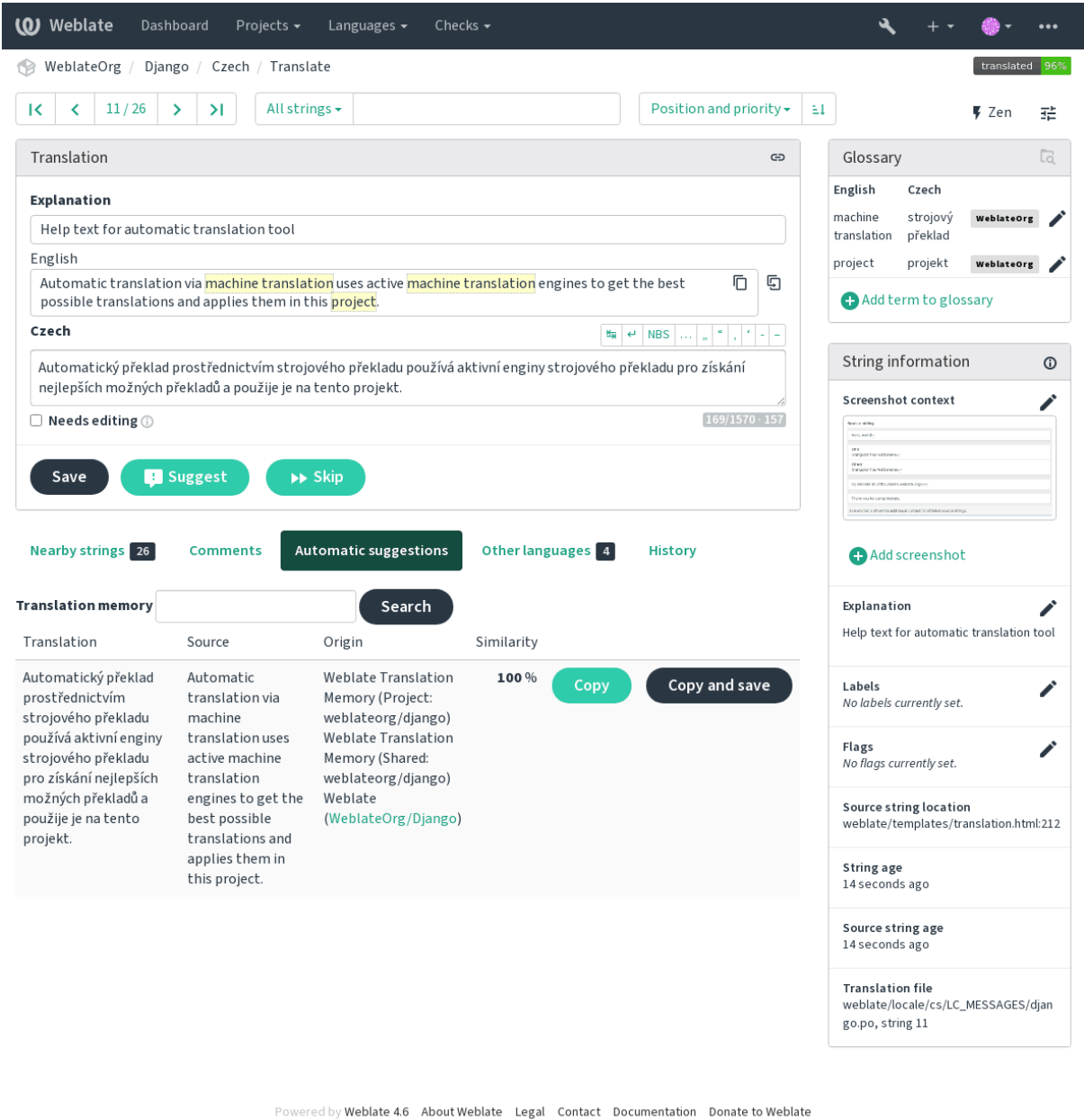
Use the explanation to clarify scope or usage of the translation. You can use Markdown to include links and other markup.

Contexto visual para textos

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:



In addition to *Informações adicionais sobre textos 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 ▾

+

▾

...

WebOrg / Django / Screenshots / Automatic translation

Screenshot has been uploaded, you can now assign it to source strings.

Assigned source strings

English	Location	Assigned screenshots	Actions
No matching strings found.			
Screenshot is shown to add visual context for all listed source strings.			

Assign source strings

English	Location	Assigned screenshots	Actions
Loading results...			

Source string search

Search

Automatically recognize

Image

Source string

Hello, world! ⁺

One
Orangutan has %d banana. ⁺

Other
Orangutan has %d bananas. ⁺

Try Weblate at <http://demo.weblate.org/>! ⁺

Thank you for using Weblate.

Screenshot is shown to add visual context for all listed source strings.

Edit screenshot

Screenshot name

Automatic translation

Image

Currently: [screenshots/screenshot.png](#)
Change:

Choose File

 No file chosen
Upload JPEG or PNG images up to 2000x2000 pixels.

Save

Screenshot details

Created	now
Uploaded by	testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

Delete

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

2.11. Processo de tradução

265

2.12 Verificações e correções

2.12.1 Correções automáticas personalizadas

Você também pode implementar sua própria correção automática, além das padrão e incluí-las em `AUTOFIX_LIST`.

As correções automáticas são poderosas, mas também podem causar danos; tenha cuidado ao escrever um.

Por exemplo, a correção automática a seguir iria substituir cada ocorrência do texto `foo`, em uma 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 para a classe Python em `AUTOFIX_LIST`, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

2.12.2 Personalizando comportamento usando marcadores

Você pode ajustar o comportamento de Weblate (principalmente de verificações) para cada texto fonte (na revisão de textos fontes, veja *Informações adicionais sobre textos fonte*) ou em *Configuração de componente* (*Marcadores de tradução*). Alguns formatos de arquivo também permitem especificar marcadores diretamente no formato (veja *Formatos de arquivos suportados*).

Os marcadores são separados por vírgulas, os parâmetros são separados por caracteres de dois pontos. Você pode usar aspas para incluir espaços em branco ou caracteres especiais no texto. Por exemplo:

```
placeholders:"special:value":"other value", regex:.*
```

Aqui está uma lista de marcadores atualmente aceitos:

rst-text Trata um texto como um documento reStructuredText, afeta *Tradução não alterada*.

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 O texto deve consistir apenas em uma URL.

safe-html O texto deve fazer uso seguro de HTML, veja *HTML inseguro*.

read-only A texto é somente leitura e não deve ser editado no Weblate, veja *Textos somente leitura*.

priority:N Prioridade do texto. Os textos de maior prioridade são apresentados primeiro para tradução. A prioridade padrão é 100, a maior prioridade que um texto tem, mais cedo é oferecido para tradução.

max-length:N Limita o comprimento máximo de um texto a caracteres N, veja *Comprimento máximo da tradução*.

xml-text Trata o texto como documento XML, afeta *Sintaxe XML* e *Marcação XML*.

font-family:NOME Define a família de fontes para verificações da renderização, veja *Gerenciando fontes*.

font-weight:PESO Define o peso da fonte para verificações da renderização, veja *Gerenciando fontes*.

font-size:TAMANHO Define o tamanho da fonte para verificações da renderização, veja *Gerenciando fontes*.

font-spacing:ESPAÇAMENTO Define letter spacing for rendering checks, see *Gerenciando fontes*.

placeholders:NAME:NAME2:... Textos de espaço reservado esperados na tradução, veja *Espaços reservados*.

replacements:DE:PARA:DE2:PARA2... Replacements to perform when checking resulting text parameters (for example in *Tamanho máximo da tradução* or *Comprimento máximo da tradução*). The typical use case for this is to expand placeables to ensure that the text fits even with long values, for example: `replacements:%s:"John Doe"`.

variants:SOURCE Mark this string as a variant of string with matching source. See variants.

regex:EXPRESSÃO REGULAR Expressão regular para corresponder à tradução; veja *Expressão regular*.

forbidden Indicates forbidden translation in a glossary, see *Traduções proibidas*.

python-format, c-format, php-format, python-brace-format, javascript-format, c-sharp-format, java-format Treats all strings like format strings, affects *Textos formatados*, *Textos formatados*, *Textos formatados*, *Textos formatados*, *Textos formatados*, *Textos formatados*, *Textos formatados*, *Textos formatados*, *Textos formatados*, *Textos formatados*, *Tradução não alterada*.

strict-same Faz com que “Tradução não alterada” evite usar a lista negra de palavras embutidas, veja *Tradução não alterada*.

check-glossary Habilita a verificação de qualidade “Não seguir glossário”.

ignore-bbcode Ignora a verificação de qualidade “Marcação de BBcode”.

ignore-duplicate Ignora a verificação de qualidade “Palavras consecutivas duplicadas”.

ignore-check-glossary Skip the “Does not follow glossary” quality check.

ignore-double-space Ignora a verificação de qualidade “Espaço duplo”.

ignore-angularjs-format Ignora a verificação de qualidade “Texto de interpolação AngularJS”.

ignore-c-format Ignora a verificação de qualidade “Formato C”.

ignore-c-sharp-format Ignora a verificação de qualidade “Formato C#”.

ignore-es-format Ignora a verificação de qualidade “Literais de modelo de ECMAScript”.

ignore-i18next-interpolation Ignora a verificação de qualidade “Interpolação de i18next”.

ignore-java-format Ignora a verificação de qualidade “Formato Java”.

ignore-java-messageformat Ignora a verificação de qualidade “MessageFormat do Java”.

ignore-javascript-format Ignora a verificação de qualidade “Formato JavaScript”.

ignore-lua-format Skip the “Lua format” quality check.

ignore-percent-placeholders Ignora a verificação de qualidade “Espaços reservados de porcentagem”.

ignore-perl-format Ignora a verificação de qualidade “Formato Perl”.

ignore-php-format Ignora a verificação de qualidade “Formato PHP”.

ignore-python-brace-format Ignora a verificação de qualidade “Formato de chaves Python”.

ignore-python-format Ignora a verificação de qualidade “Formato Python”.

ignore-qt-format Ignora a verificação de qualidade “Formato Qt”.

ignore-qt-plural-format Ignora a verificação de qualidade “Formato de plural Qt”.

ignore-ruby-format Ignora a verificação de qualidade “Formato Ruby”.

ignore-vue-format Ignora a verificação de qualidade “Formatação Vue I18n”.

ignore-translated Ignora a verificação de qualidade “Foi traduzido”.

ignore-inconsistent Ignora a verificação de qualidade “Inconsistente”.

ignore-kashida Ignora a verificação de qualidade “Letra Kashida usada”.

ignore-md-link Ignora a verificação de qualidade “Links Markdown”.

ignore-md-reflink Ignora a verificação de qualidade “Referências Markdown”.

ignore-md-syntax Ignora a verificação de qualidade “Sintaxe Markdown”.

ignore-max-length Ignora a verificação de qualidade “Comprimento máximo da tradução”.

ignore-max-size Ignora 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 Ignora a verificação de qualidade “Reticências não correspondentes”.

ignore-end-exclamation Ignora a verificação de qualidade “Ponto de exclamação não correspondente”.

ignore-end-stop Ignora a verificação de qualidade “Ponto final não correspondente”.

ignore-end-question Ignora a verificação de qualidade “Ponto de interrogação não correspondente”.

ignore-end-semicolon Ignora a verificação de qualidade “ponto e vírgula não correspondente”.

ignore-newline-count Ignora a verificação de qualidade “Quebras de linha descasadas”.

ignore-plurals Ignora a verificação de qualidade “Faltam plurais”.

ignore-placeholders Ignora a verificação de qualidade “Espaços reservados”.

ignore-punctuation-spacing Ignora a verificação de qualidade “Espaçamento de pontuação”.

ignore-regex Ignora a verificação de qualidade “Expressão regular”.

ignore-same-plurals Ignora a verificação de qualidade “Mesmos plurais”.

ignore-begin-newline Ignora a verificação de qualidade “Nova linha no início”.

ignore-begin-space Ignora a verificação de qualidade “Espaços no início”.

ignore-end-newline Ignora a verificação de qualidade “Nova linha no final”.

ignore-end-space Ignora a verificação de qualidade “Espaço no final”.

ignore-same Ignora a verificação de qualidade “Tradução não alterada”.

ignore-safe-html Ignora a verificação de qualidade “HTML inseguro”.

ignore-url Ignora a verificação de qualidade “URL”.

ignore-xml-tags Ignora a verificação de qualidade “Marcação XML”.

ignore-xml-invalid Ignora a verificação de qualidade “Sintaxe XML”.

ignore-zero-width-space Ignora a verificação de qualidade “Espaço com largura zero”.

ignore-ellipsis Ignora a verificação de qualidade “Reticências”.

ignore-long-untranslated Ignora a verificação de qualidade “Não traduzido a muito tempo”.

ignore-multiple-failures Ignora a verificação de qualidade “Várias verificações com falha”.

ignore-unnamed-format Ignora a verificação de qualidade “Várias variáveis sem nome”.

ignore-optional-plural Ignora a verificação de qualidade “Não pluralizado”.

Nota: Geralmente, a regra é chamada de `ignore-*` para qualquer verificação, usando seu identificador, para que você possa usá-la mesmo para suas verificações personalizadas.

Esses marcadores são entendidos tanto nas configurações de *Configuração de componente*, por configurações de textos fonte quanto no próprio arquivo de tradução (por exemplo, no GNU gettext).

2.12.3 Forçando verificações

Novo na versão 3.11.

Você pode configurar uma lista de verificações que não podem ser ignoradas definindo *Verificações forçadas* em *Configuração de componente*. Cada verificação listada não pode ser ignorada na interface do usuário e qualquer texto com falha nesta verificação é marcado como *Precisa de edição* (veja *Estados de tradução*).

2.12.4 Gerenciando fontes


Novo na versão 3.7.

Dica: Fonts uploaded into Weblate are used purely for purposes of the *Tamanho máximo da tradução* check, they do not have an effect in Weblate user interface.

A verificação *Tamanho máximo da tradução* usada para calcular as dimensões do texto renderizado precisa ser carregada no Weblate e selecionada usando um sinalizador de tradução (ver *Personalizando comportamento usando marcadores*).

Ferramenta de gerenciamento de fontes do Weblate em *Fonts* sob o menu *Gerenciar* do seu projeto de tradução fornece interface para carregar e gerenciar fontes. As fontes TrueType ou OpenType podem ser carregadas, configurar grupos de fontes e usá-los na verificação.

Os grupos de fontes permitem definir diferentes fontes para diferentes idiomas, 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](#)

O grupos de fontes 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 4.6 About Weblate Legal Contact Documentation Donate to Weblate

A família de fontes e o estilo são automaticamente reconhecidos após carregá-los:

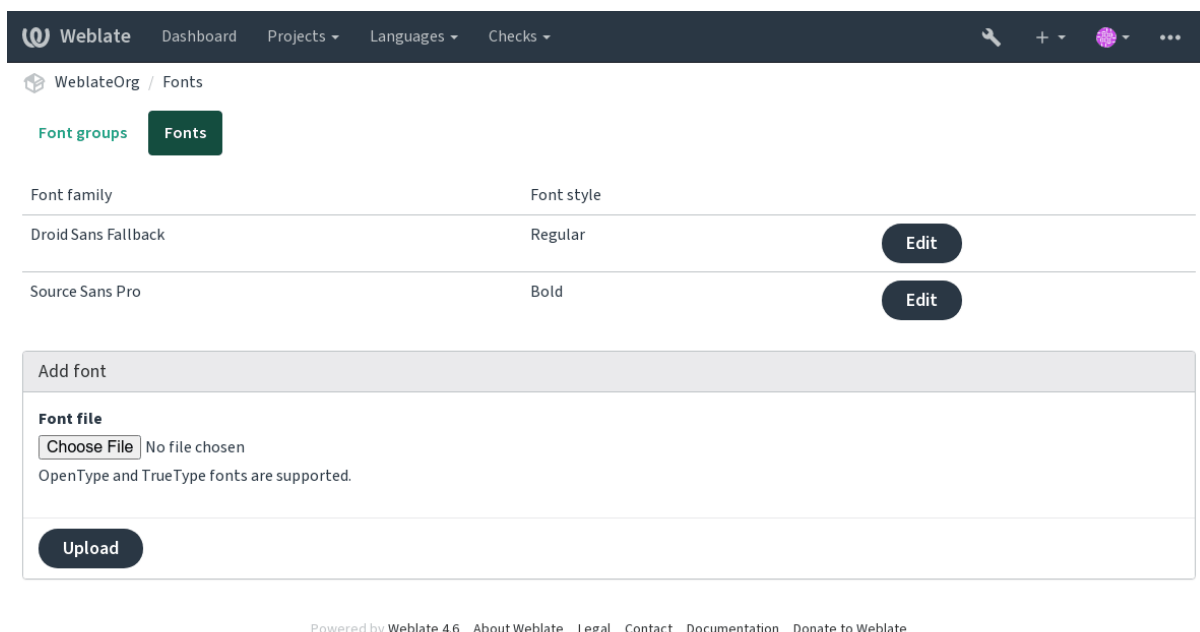
 Weblate Dashboard Projects Languages Checks

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 4.6 About Weblate Legal Contact Documentation Donate to Weblate

Você pode ter um número de fontes carregadas para Weblate:



Para usar as fontes para verificar o comprimento do texto, passe-o os marcadores apropriados (veja [Personalizando comportamento usando marcadores](#)). Você provavelmente precisará dos seguintes:

max-size:500 Define o máximo de largura.

font-family:ubuntu Define o grupo de fontes para usar especificando seu identificador.

font-size:22 Define o tamanho da fonte.

2.12.5 Escrevendo as próprias verificações

Uma ampla gama de verificações de qualidade estão embutidas, (veja [Verificações de qualidade](#)), embora eles possam não cobrir tudo o que você deseja verificar. A lista de verificações realizadas pode ser ajustada usando [CHECK_LIST](#) e você também pode adicionar verificações personalizadas.

1. Crie uma subclasse de `weblate.checks.Check`
2. Defina alguns atributos.
3. Implemente o método `check` (se você quiser lidar com plurais em seu código) ou o método `check_single` (que faz isso por você).

Alguns exemplos:

Para instalar verificações personalizadas, forneça um caminho totalmente qualificado para a classe Python em [CHECK_LIST](#), veja [Verificações de qualidade personalizadas, extensões e correções automáticas](#).

Verificando se o texto de tradução não contém “foo”

Esta é uma verificação bastante simples que apenas verifica se a tradução não possui o texto “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
```

(continua na próxima página)

(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
```

(continua na próxima página)

```
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 de máquina

Built-in support for several machine translation services and can be turned on by the administrator using `MT_SERVICES` for each one. They come subject to their terms of use, so ensure you are allowed to use them how you want.

O idioma fonte pode ser configurado na *Configuração de projeto*.

2.13.1 amaGama

Special installation of *tmserver* run by the authors of Virtaal.

Turn on this service by adding `weblate.machinery.tmserver.AmagamaTranslation` to `MT_SERVICES`.

Ver também:

Installing amaGama, Amagama, amaGama Translation Memory

2.13.2 Apertium

A libre software machine translation platform providing translations to a limited set of languages.

The recommended way to use Apertium is to run your own Apertium-APy server.

Turn on this service by adding `weblate.machinery.apertium.ApertiumAPYTranslation` to `MT_SERVICES` and set `MT_APERTIUM_APY`.

Ver também:

`MT_APERTIUM_APY`, Apertium website, Apertium APy documentation

2.13.3 AWS

Novo na versão 3.1.

Amazon Translate is a neural machine translation service for translating text to and from English across a breadth of supported languages.

1. Turn on this service by adding `weblate.machinery.aws.AWSTranslation` to `MT_SERVICES`.
2. Install the *boto3* module.
3. Configure Weblate.

Ver também:

`MT_AWS_REGION`, `MT_AWS_ACCESS_KEY_ID`, `MT_AWS_SECRET_ACCESS_KEY` [Amazon Translate Documentation](#)

2.13.4 Baidu API machine translation

Novo na versão 3.2.

Machine translation service provided by Baidu.

This service uses an API and you need to obtain an ID and API key from Baidu to use it.

Turn on this service by adding `weblate.machinery.baidu.BaiduTranslation` to `MT_SERVICES` and set `MT_BAIDU_ID` and `MT_BAIDU_SECRET`.

Ver também:

`MT_BAIDU_ID`, `MT_BAIDU_SECRET` [Baidu Translate API](#)

2.13.5 DeepL

Novo na versão 2.20.

DeepL is paid service providing good machine translation for a few languages. You need to purchase *DeepL API* subscription or you can use legacy *DeepL Pro (classic)* plan.

Turn on this service by adding `weblate.machinery.deepl.DeepLTranslation` to `MT_SERVICES` and set `MT_DEEPL_KEY`.

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). You can toggle this by `MT_DEEPL_API_VERSION`.

Ver também:

`MT_DEEPL_KEY`, `MT_DEEPL_API_VERSION`, [DeepL website](#), [DeepL pricing](#), [DeepL API documentation](#)

2.13.6 Glosbe

Free dictionary and translation memory for almost every living language.

The API is gratis to use, but subject to the used data source license. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Turn on this service by adding `weblate.machinery.glosbe.GlosbeTranslation` to `MT_SERVICES`.

Ver também:

[Glosbe website](#)

2.13.7 Google Translate

Machine translation service provided by Google.

This service uses the Google Translation API, and you need to obtain an API key and turn on billing in the Google API console.

To turn on this service, add `weblate.machinery.google.GoogleTranslation` to `MT_SERVICES` and set `MT_GOOGLE_KEY`.

Ver também:

`MT_GOOGLE_KEY`, [Google translate documentation](#)

2.13.8 Google Translate API V3 (Advanced)

Machine translation service provided by Google Cloud services.

This service differs from the former one in how it authenticates. To enable service, add `weblate.machinery.google.v3.GoogleV3Translation` to `MT_SERVICES` and set

- `MT_GOOGLE_CREDENTIALS`
- `MT_GOOGLE_PROJECT`

If `location` fails, you may also need to specify `MT_GOOGLE_LOCATION`.

Ver também:

`MT_GOOGLE_CREDENTIALS`, `MT_GOOGLE_PROJECT`, `MT_GOOGLE_LOCATION` [Google translate documentation](#)

2.13.9 Microsoft Cognitive Services Translator

Novo na versão 2.10.

Machine translation service provided by Microsoft in Azure portal as a one of Cognitive Services.

Weblate implements Translator API V3.

To enable this service, add `weblate.machinery.microsoft.MicrosoftCognitiveTranslation` to `MT_SERVICES` and set `MT_MICROSOFT_COGNITIVE_KEY`.

Translator Text API V2

The key you use with Translator API V2 can be used with API 3.

Translator Text API V3

You need to register at Azure portal and use the key you obtain there. With new Azure keys, you also need to set `MT_MICROSOFT_REGION` to locale of your service.

Ver também:

`MT_MICROSOFT_COGNITIVE_KEY`, `MT_MICROSOFT_REGION`, [Cognitive Services - Text Translation API](#), [Microsoft Azure Portal](#)

2.13.10 Microsoft Terminology Service

Novo na versão 2.19.

The Microsoft Terminology Service API allows you to programmatically access the terminology, definitions and user interface (UI) strings available in the Language Portal through a web service.

Turn this service on by adding `weblate.machinery.microsoftterminology.MicrosoftTerminologyService` to `MT_SERVICES`.

Ver também:

Microsoft Terminology Service API

2.13.11 ModernMT

Novo na versão 4.2.

Turn this service on by adding `weblate.machinery.modernmt.ModernMTTranslation` to `MT_SERVICES` and configure `MT_MODERNMT_KEY`.

Ver também:

ModernMT API, `MT_MODERNMT_KEY`, `MT_MODERNMT_URL`

2.13.12 MyMemory

Huge translation memory with machine translation.

Free, anonymous usage is currently limited to 100 requests/day, or to 1000 requests/day when you provide a contact e-mail address in `MT_MYMEMORY_EMAIL`. You can also ask them for more.

Turn on this service by adding `weblate.machinery.mymemory.MyMemoryTranslation` to `MT_SERVICES` and set `MT_MYMEMORY_EMAIL`.

Ver também:

`MT_MYMEMORY_EMAIL`, `MT_MYMEMORY_USER`, `MT_MYMEMORY_KEY`, MyMemory website

2.13.13 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`.

Ver também:

`MT_NETEASE_KEY`, `MT_NETEASE_SECRET` Netease Sight Translation Platform

2.13.14 tmserver

You can run your own translation memory server by using the one bundled with Translate-toolkit and let Weblate talk to it. You can also use it with an amaGama server, which is an enhanced version of tmserver.

1. 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/"
```

Ver também:

[MT_TMSERVER](#), [tmserver](#) [Installing amaGama](#), [Amagama](#), [Amagama Translation Memory](#)

2.13.15 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`.

Ver também:

[MT_YANDEX_KEY](#), [Yandex Translate API](#), [Powered by Yandex.Translate](#)

2.13.16 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`.

Ver também:

[MT_YOUDAO_ID](#), [MT_YOUDAO_SECRET](#) [Youdao Zhiyun Natural Language Translation Service](#)

2.13.17 Weblate

Weblate can be the source of machine translations as well. It is based on the Woosh fulltext engine, and provides both exact and inexact matches.

Turn on these services by adding `weblate.machinery.weblatetm.WeblateTranslation` to `MT_SERVICES`.

2.13.18 Weblate Translation Memory

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.

2.13.19 SAP Translation Hub

Machine translation service provided by SAP.

You need to have a SAP account (and enabled the SAP Translation Hub in the SAP Cloud Platform) to use this service.

Turn on this service by adding `weblate.machinery.saptranslationhub.SAPTranslationHub` to `MT_SERVICES` and set the appropriate access to either sandbox or the productive API.

Nota: To access the Sandbox API, you need to set `MT_SAP_BASE_URL` and `MT_SAP_SANDBOX_APIKEY`.

To access the productive API, you need to set `MT_SAP_BASE_URL`, `MT_SAP_USERNAME` and `MT_SAP_PASSWORD`.

Ver também:

`MT_SAP_BASE_URL`, `MT_SAP_SANDBOX_APIKEY`, `MT_SAP_USERNAME`, `MT_SAP_PASSWORD`, `MT_SAP_USE_MT` SAP Translation Hub API

2.13.20 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
```

(continua na próxima página)

(continuação da página anterior)

```
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation

class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

    def download_languages(self):
        """Return list of languages your machine translation supports."""
        return {"cs"}


    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```


You can list own class in `MT_SERVICES` and Weblate will start using that.

2.14 Extensões

Novo na versão 2.19.

Extensões fornecem maneiras para personalizar e automatizar o fluxo de trabalho de tradução. Administradores podem gerenciar extensões a partir do menu *Gerenciar* ↓ *Extensões* dos respectivos componente de tradução.



[Weblate](#)
[Dashboard](#)
[Projects ▾](#)
[Languages ▾](#)
[Checks ▾](#)
🔧
+
🌐
⋮


[WeblateOrg](#) /
 [Language names](#) /
 Addons

Installed addons ⓘ


There are no addons currently installed.

Available addons ⓘ


Automatic translation ⓘ

Automatically translates strings using machine translation or other components.


Install


Add missing languages ⓘ

Ensures a consistent set of languages is used for all components within a project.

project wide


Install


Component discovery ⓘ

Automatically adds or removes project components based on file changes in the version control system.


repository wide

Install


Bulk edit ⓘ


Bulk edit flags, labels, or states of strings.

Install


Statistics generator ⓘ


Generates a file containing detailed info about the translation status.

Install


Pseudolocale generation ⓘ


Generates a translation by adding prefix and suffix to source strings automatically.

Install


Contributors in comment ⓘ


Updates the comment part of the PO file header to include contributor names and years of contributions.

Install


Customize gettext output ⓘ


Allows customization of gettext output behavior, for example line wrapping.

Install


Generate MO files ⓘ


Automatically generates a MO file for every changed PO file.

Install


Update PO files to match POT (msgmerge) ⓘ

Updates all PO files (as configured by "Filemask") to match the POT file (as configured by "Template for new translations") using msgmerge.


Install


Squash Git commits ⓘ

Squash Git commits prior to pushing changes.

repository wide


Install


Stale comment removal ⓘ

Set a timeframe for removal of comments.

project wide

Install


Stale suggestion removal ⓘ

Set a timeframe for removal of suggestions.

project wide

Install

Some addons will ask for additional configuration during installation.

2.14.1 Extensões embutidas

Tradução automática

Novo na versão 3.9.

Traduz automaticamente textos usando tradução de máquina ou outros componentes.

Ela é acionada:

- Quando novos textos aparecem em um componente.
- Once in a month for every component, this can be configured using `BACKGROUND_TASKS`.

Ver também:

Tradução automática, Mantendo traduções iguais entre componentes

CDN de localização JavaScript

Novo na versão 4.2.

Publica traduções para rede de entrega de conteúdo (CDN) para uso em localização JavaScript ou HTML.

Pode ser usado para localizar páginas HTML estáticas ou para carregar a localização no código JavaScript.

Gera um URL exclusivo para seu componente, que pode ser incluído em páginas HTML para localizá-los. Veja `weblate-cdn` para mais detalhes.

Ver também:

`cdn-addon-config`, `weblate-cdn`, `cdn-addon-extract`, `cdn-addon-html`

Remover textos vazios

Novo na versão 4.4.

Remove textos sem uma tradução dos arquivos de tradução.

Use isso para não ter textos vazios em arquivos de tradução (por exemplo, se sua biblioteca de localização os exibe como em falta em vez de recorrer ao texto fonte).

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Limpar arquivos de tradução

Atualiza todos os arquivos de tradução para corresponder ao arquivo base monolíngue. Para a maioria dos formatos isto significa remover chaves de tradução obsoletas que não estão mais presentes no arquivo base.

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Adiciona idiomas em falta

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 automaticamente os textos recém-adicionadas com *Tradução automática*.

Descoberta de componente

Adiciona ou remove automaticamente componentes do projeto com base nas alterações de arquivo no sistema de controle de versão.

Acionada toda vez que o VCS é atualizado, de outra forma semelhante ao comando de gerenciamento *import_project*. Desta forma, você pode rastrear vários componentes de tradução dentro de um VCS.

A correspondência é feita usando expressões regulares que permitem uma configuração complexa, mas é necessário algum conhecimento para fazê-la. Alguns exemplos para casos de uso comum podem ser encontrados na seção de ajuda de extensões.

Uma vez que você acertar *Salvar*, uma prévia dos componentes correspondentes será apresentada, de onde você pode verificar se a configuração realmente corresponde às suas necessidades:

Weblate

Dashboard

Projects

Languages

Checks

+

WebOrg

Language names

Addons

Component discovery

Configure addon

Please review and confirm the matched components.

Component	Matched files
Following components would be created	
Djangojs	<div>weblate/locale/cs/LC_MESSAGES/djangojs.po (cs)</div> <div>weblate/locale/hu/LC_MESSAGES/djangojs.po (hu)</div> <div>weblate/locale/he/LC_MESSAGES/djangojs.po (he)</div>
Django	<div>weblate/locale/cs/LC_MESSAGES/django.po (cs)</div> <div>weblate/locale/hu/LC_MESSAGES/django.po (hu)</div> <div>weblate/locale/he/LC_MESSAGES/django.po (he)</div>

☐ I confirm the above matches look correct

Regular expression to match translation files against

weblate/locale/(?P<language>[^\.]*)/LC_MESSAGES/(?P<component>[^\.]*).po

File format

gettext PO file

Customize the component name

{{ component|title }}

Define the monolingual base filename

Leave empty for bilingual translation files.

Define the base file for new translations

weblate/locale/{{ component }}.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Language filter

^(cs|he|hu)\$

Regular expression to filter translation files against when scanning for filemask.

☒ Clone addons from the main component to the newly created ones

☐ Remove components for inexistant files

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description
(?P<language>[^\.]*)/(?P<component>[^\.]*).po	cs/application.po cs/website.po de/application.po de/website.po	One folder per language containing translation files for components.
locale/(?P<language>[^\.]*)/LC_MESSAGES/(?P<component>[^\.]*).po	locale/cs/LC_MESSAGES/application.po locale/cs/LC_MESSAGES/website.po locale/de/LC_MESSAGES/application.po locale/de/LC_MESSAGES/website.po	Usual structure for storing gettext PO files.
src/locale/(?P<component>[^\.]*).(?P<language>[^\.]*).po	src/locale/application.cs.po src/locale/website.cs.po src/locale/application.de.po src/locale/website.de.po	Using both component and language name within filename.
locale/(?P<language>[^\.]*)/(?P<component>[^\.]*)(?P=language).po	locale/cs/application/cs.po locale/cs/website/cs.po locale/de/application/de.po locale/de/website/de.po	Using language in both path and filename.
res/values-(?P<language>[^\.]*)/strings-(?P<component>[^\.]*).xml	res/values-cs/strings-about.xml res/values-cs/strings-help.xml res/values-de/strings-about.xml res/values-de/strings-help.xml	Android resource strings, split into several files.

You can use Django template markup in both component name and the monolingual base filename, for example:

{{ component }}

Component filename match

{{ component|title }}

Component filename with upper case first letter

Save

Dica: Component discovery addon uses *URLs internas do Weblate*. It's a convenient way to share VCS setup between multiple components. Linked components use the local repository of the main component set up by filling `weblate:///project/main-component` into the *Repositório do código-fonte* field (in *Manage* ↓ *Settings* ↓ *Version control system*) of each respective component. This saves time with configuration and system resources too.

Ver também:

Template markup

Editor em massa

Novo na versão 3.11.

Edição em massa de marcadores, etiquetas ou estados de cadeias.

Automatize o etiquetamento começando com a consulta de pesquisa `NOT has:label` e adicione etiquetas até que todas os textos tenham todas as etiquetas necessárias. Outras operações automatizadas para metadados Weblate também podem ser feitas.

Exemplos:

Tabela 4: Label new strings automatically

Consulta de pesquisa	<code>NOT has:label</code>
Etiquetas a adicionar	<code>recente</code>

Tabela 5: Marcar como somente leitura todas as entradas do changelog de Arquivos de metadados de loja de aplicativos

Consulta de pesquisa	<code>language:en AND key:changelogs/</code>
Marcadores de tradução a adicionar	<code>read-only</code>

Ver também:

Editor em massa, *Personalizando comportamento usando marcadores*, *labels*

Marcar traduções não alteradas como “Necessita edição”

Novo na versão 3.1.

Sempre que um novo texto traduzível é importado do VCS e corresponde a um texto fonte, é sinalizado como necessitando de edição no Weblate. Especialmente útil para formatos de arquivo que incluem textos fonte para textos não traduzidas.

Dica: You might also want to tighten the *Tradução não alterada* check by adding `strict-same` flag to *Marcadores de tradução*.

Ver também:

Estados de tradução

Marcar novos textos fonte como “Necessita edição”

Sempre que um novo texto fonte for importado do VCS, ele é marcado como necessita edição no Weblate. Desta forma você pode filtrar facilmente e editar os textos fontes escritos pelos desenvolvedores.

Ver também:

Estados de tradução

Marcar novas traduções como “Necessita edição”

Sempre que um novo texto traduzível for importado do VCS, ele é marcado como necessita edição no Weblate. Desta forma você pode filtrar facilmente e editar as traduções criadas pelos desenvolvedores.

Ver também:

Estados de tradução

Gerador de estatísticas

Gera um arquivo contendo as informações detalhadas sobre o status da tradução.

Você pode usar um modelo do Django, tanto de nome de arquivo e conteúdo, veja [Markdown](#) para uma descrição detalhada de marcação.

Por exemplo, a geração de um arquivo de resumo para cada tradução:

Nome do arquivo 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 }}"
}
```

Ver também:

Template markup

Geração de pseudolocalidade

Gera uma tradução adicionando prefixo e sufixo de textos fonte automaticamente.

Pseudolocales are useful to find strings that are not prepared for localization. This is done by altering all translatable source strings to make it easy to spot unaltered strings when running the application in the pseudolocale language.

Finding strings whose localized counterparts might not fit the layout is also possible.

Dica: You can use real languages for testing, but there are dedicated pseudolocales available in Weblate - *en_XA* and *ar_XB*.

Colaboradores nos comentários

Atualiza a parte do comentário do cabeçalho do arquivo PO para incluir nomes de contribuidores e anos de contribuições.

O cabeçalho do arquivo 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 variável ALL_LINGUAS no arquivo “configure”

Atualiza a variável ALL_LINGUAS em arquivos `configure`, `configure.in` ou `configure.ac`, quando uma nova tradução é adicionada.

Personalizar saída do gettext

Permite personalizar o comportamento de saída do gettext, por exemplo as quebra de linhas.

Ela oferece as seguintes opções:

- Quebrar linhas em 77 caracteres e em novas linhas
- Quebrar linhas apenas em 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 arquivo LINGUAS

Atualiza o arquivo LINGUAS quando uma nova tradução for adicionada.

Gerar arquivos MO

Gera automaticamente um arquivo MO para cada arquivo PO alterado.

The location of the generated MO file can be customized and the field for it uses *Template markup*.

Atualizar arquivos PO para corresponder ao POT (msgmerge)

Atualiza todos os arquivos PO (configurados por *Máscara de arquivo*) para corresponder ao arquivo POT (configurado por *Modelo para novas traduções*) usando `msgmerge`.

Acionada sempre que novas alterações forem trazidas do repositório upstream. A maioria das opções de linha de comando do msgmerge podem ser definidas por meio da configuração da extensão.

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Squash de commits git

Fazer squash de commits Git antes de fazer push das alterações.

Commits Git podem ter squash feito antes de fazer o push das alterações em um dos seguintes modos:

Novo na versão 3.4.

- Todos os commits em um só
- Por idioma
- Por arquivo

Novo na versão 3.5.

- Por autor

As mensagens de compromisso originais são mantidas, mas a autoria é perdida a menos que *Por autor* seja selecionada ou a mensagem de compromisso 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 saída JSON

Permite ajustar o comportamento de saída do JSON, por exemplo recuo ou classificação.

Formatar o arquivo de propriedades Java

Ordena o arquivo de propriedades Java.

Remoção de comentário obsoleto

Novo na versão 3.7.

Define um período de tempo até 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 sua importância.

Remoção de sugestão obsoleta

Novo na versão 3.7.

Define um período de tempo até a remoção de sugestões.

Pode ser muito útil em relação à votação em sugestão (veja [Revisão por pares](#)) para remover sugestões que não recebem votos positivos suficientes em um determinado período de tempo.

Atualizar arquivos RESX

Novo na versão 3.9.

Atualiza todos os arquivos de tradução para corresponder ao arquivo base monolíngue do upstream. Textos não usados são removidos e novos são adicionados como cópias do texto fonte.

Dica: Use *Limpar arquivos de tradução* se você só quiser remover chaves de tradução obsoletas.

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Personalizar saída YAML

Novo na versão 3.10.2.

Permite ajustar o comportamento de saída do YAML, por exemplo comprimento de linha e 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

Você pode escrever suas próprias extensões também. 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.

Ver 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 você tem que escrever algum código para embrulhar 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
# 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/>.
#
```

(continua na próxima página)

```
"""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*.

O script é executado com o diretório atual definido para a raiz do repositório VCS para qualquer componente.

Além disso, estão disponíveis as seguintes variáveis de ambiente:

WL_VCS

Sistema de controle de versão usado.

WL_REPO

URL do repositório upstream.

WL_PATH

Caminho absoluto para o repositório VCS.

WL_BRANCH

Novo na versão 2.11.

Ramo do repositório configurado no componente atual.

WL_FILEMASK

Máscara de arquivo para componente atual.

WL_TEMPLATE

Nome de arquivo de modelo para traduções monolíngues (pode estar vazio).

WL_NEW_BASE

Novo na versão 2.14.

Nome de arquivo do arquivo usado para criar novas traduções (pode estar vazio).

WL_FILE_FORMAT

Formato de arquivo 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.

Ver também:

Configuração de componente

Processamento de repositório pós-atualização

Pode ser usado para atualizar arquivos de tradução quando a fonte VCS do upstream alterar. Para conseguir isso, lembre-se que o Weblate só vê arquivos dos quais se fez commit com o VCS, então você precisa fazer commit das alterações como parte do script.

Por exemplo, com Gulp, você pode fazê-lo usando o seguinte código:

```
#!/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.

Ele 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.

Ver também:

Esquema de memória de tradução do Weblate

Interface de gerenciamento

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 arquivos 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, você precisa reiniciar o Weblate - tanto os processos WSGI quanto os Celery.

In case it is run as `mod_wsgi`, you need to restart Apache to reload the configuration.

Ver também:

Verifique também a [documentação do Django](#) para parâmetros de configuração do próprio Django.

2.16.1 AKISMET_API_KEY

O Weblate pode usar o Akismet para verificar sugestões recebidas anonimamente para spam. Visite akismet.com para comprar uma chave API e associá-la a um site.

2.16.2 ANONYMOUS_USER_NAME

O nome de usuário dos usuários não autenticados.

Ver também:

Controle de acesso

2.16.3 AUDITLOG_EXPIRY

Novo na versão 3.6.

Quantos dias o Weblate deve manter registros de auditoria, que contêm informações sobre a atividade da conta.

O padrão é 180 dias.

2.16.4 AUTH_LOCK_ATTEMPTS

Novo na versão 2.14.

Número máximo de tentativas de autenticação com falha antes da aplicação da limitação de taxa.

Atualmente, isso é aplicado nos seguintes locais:

- Logins. Exclui a senha da conta, impedindo que o usuário entre sem solicitar uma nova senha.
- Redefinições de senha. Impede que novos e-mails sejam enviados, evitando o envio de spam aos usuários com muitas tentativas de redefinição de senha.

O padrão é 10.

Ver 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 textos são aceitos.

Atualiza todos repositórios diariamente.

Dica: Útil se você não estiver usando *Ganchos de notificação* para atualizar automaticamente os repositórios do Weblate.

Nota: Existem opções de ligar/desligar, além da seleção de textos para compatibilidade com versões anteriores.

As opções são:

"none" Sem atualizações diárias.

"remote" e também False Atualiza apenas os repositórios remotos.

"full" e também True Atualiza os repositórios remotos e mescla os cópia de trabalho.

Nota: Isso requer que *Tarefas de fundo usando Celery* esteja funcionando 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 seguintes serviços funcionam:

Gravatar (padrão), conforme <https://gravatar.com/> `AVATAR_URL_PREFIX` = `'https://www.gravatar.com/'`

Libravatar, conforme <https://www.libravatar.org/> `AVATAR_URL_PREFIX` = `'https://www.libravatar.org/'`

Ver 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 senha temporária dos e-mails de redefinição de senha são válidos. Definido em número de segundos, usando como padrão 172800 (2 dias).

2.16.8 AUTH_PASSWORD_DAYS

Novo na versão 2.15.

Quantos dias usando a mesma senha deve ser permitido.

Nota: Mudanças de senha feitas anteriormente ao Weblate 2.15 não serão consideradas para essa política.

O padrão é 180 dias.

2.16.9 AUTOFIX_LIST

Lista de correções automáticas para aplicar ao salvar um texto.

Nota: Forneça um caminho totalmente qualificado para a 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 do texto com a fonte.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis` Replaces trailing dots (...) if the source string has a corresponding ellipsis (...).

`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 dos textos sinalizados como `safe-html` (veja *HTML inseguro*).

Você pode selecionar quais usar:

```
AUTOFIX_LIST = (
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
)
```

Ver também:

Correções automáticas, Correções automáticas personalizadas

2.16.10 BACKGROUND_TASKS

Novo na versão 4.5.2.

Defines how often lengthy maintenance tasks should be triggered for a component.

Right now this controls:

- Extensão *Tradução automática*
- *Verificações e correções* recalculation

Escolhas possíveis:

- monthly (this is the default)
- weekly
- daily
- never

Nota: Increasing the frequency is not recommended when Weblate contains thousands of components.

2.16.11 BASE_DIR

Diretório base onde as fontes do Weblate estão localizadas. Usado para derivar vários outros caminhos por padrão:

- *DATA_DIR*

Valor padrão: Diretório de nível superior de fontes do Weblate.

2.16.12 BASIC_LANGUAGES

Novo na versão 4.4.

List of languages to offer users for starting new translation. When not specified built-in list is used which includes all commonly used languages, but without country specific variants.

Isso só limita usuários 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"}
```

Ver também:

Definições de idioma

2.16.13 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Customize Content-Security-Policy header for Weblate. The header is automatically generated based on enabled integrations with third-party services (Matomo, Google Analytics, Sentry, ...).

Todos esses tem como padrão uma lista vazia.

Exemplo:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Ver também:

Política de segurança de conteúdo, Políticas de Segurança de Conteúdo (CSP)

2.16.14 CHECK_LIST

Lista de verificações de qualidade para realizar em uma tradução.

Nota: Forneça um caminho totalmente qualificado para a classe Python implementando a interface de verificação.

Ajuste a lista de verificações para incluir as relevantes para você.

Todas as *Verificações de qualidade* embutidas estão ativadas por padrão, de onde você pode alterar essas configurações. Por padrão, eles são comentados em *Amostra de configuração* para que os valores padrão sejam usados. Novas verificações então realizadas para cada nova versão do Weblate.

Você pode desativar todas as verificações:

```
CHECK_LIST = ()
```

Você 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*.

Ver também:

Verificações de qualidade, *Personalizando comportamento usando marcadores*

2.16.15 COMMENT_CLEANUP_DAYS

Novo na versão 3.6.

Exclui comentários após um determinado número de dias. O padrão é `None`, ou seja, nenhuma exclusão.

2.16.16 COMMIT_PENDING_HOURS

Novo na versão 2.10.

Número de horas entre fazer o commit de alterações pendentes por meio da tarefa de segundo plano.

Ver também:

Configuração de componente, Idade das alterações para fazer commit, Executando tarefas de manutenção, `commit_pending`

2.16.17 CONTACT_FORM

Novo na versão 4.6.

Configures how e-mail from the contact form is being sent. Choose a configuration that matches your mail server configuration.

"reply-to" The sender is used in as *Reply-To*, this is the default behaviour.

"from" The sender is used in as *From*. Your mail server needs to allow sending such e-mails.

2.16.18 DATA_DIR

A pasta na qual Weblate armazena todos os dados. Ela contém links para repositórios VCS, um índice de texto e vários arquivos de configuração para ferramentas externas.

Os seguintes subdiretórios geralmente existem:

home Diretório pessoal usado para invocar scripts.

ssh Chaves e configuração de SSH.

static Default location for static Django files, specified by `STATIC_ROOT`. See *Servindo arquivos estáticos*.

media Default location for Django media files, specified by `MEDIA_ROOT`. Contains uploaded screenshots.

vcs Version control repositories for translations.

backups Dados de backup diário. Confira *Dados despejados para os backups* para detalhes.

celery Dados do agendador Celery, consulte *Tarefas de fundo usando Celery*.

fonts: User-uploaded fonts, see *Gerenciando fontes*.

Nota: Este diretório tem que ser escrito pelo Weblate. Executá-lo como uWSGI significa que o usuário `www-data` deve ter acesso de escrita.

A maneira mais fácil de conseguir isso é fazer do usuário o proprietário do diretório:

```
sudo chown www-data:www-data -R $DATA_DIR
```

O padrão é `$BASE_DIR/data`.

Ver também:

BASE_DIR, Permissões do sistema de arquivos, Fazendo backup e movendo 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"

Ver também:

Fazendo backup e movendo o Weblate

2.16.20 DEFAULT_ACCESS_CONTROL

Novo na versão 3.3.

A configuração padrão de controle de acesso para novos projetos:

0 *Público*

1 *Protegido*

100 *Privado*

200 *Personalizado*

Use *Personalizado* se você está gerenciando a ACL manualmente, o que significa não confiar no gerenciamento interno do Weblate.

Ver também:

Controle de acesso do projeto, Controle de acesso

2.16.21 DEFAULT_AUTO_WATCH

Novo na versão 4.5.

Configures whether *Automatically watch projects on contribution* should be turned on for new users. Defaults to `True`.

Ver também:

Notificações

2.16.22 DEFAULT_RESTRICTED_COMPONENT

Novo na versão 4.1.

O valor padrão para restrição de componentes.

Ver 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 padrão para diferentes operações, consulte *Configuração de componente* para detalhes.

Ver também:

Template markup, Configuração de componente, Mensagens de commit, add, merge e extensão

2.16.24 DEFAULT_ADDONS

Complementos padrão para instalar em cada componente criado.

Nota: Essa configuração afeta apenas componentes recém-criados.

Exemplo:

```
DEFAULT_ADDONS = {  
    # Addon with no parameters  
    "weblate.flags.target_edit": {},  
    # Addon with parameters  
    "weblate.autotranslate.autotranslate": {  
        "mode": "suggest",  
        "filter_type": "todo",  
        "auto_source": "mt",  
        "component": "",  
        "engines": ["weblate-translation-memory"],  
        "threshold": "80",  
    },  
}
```

Ver também:

install_addon, WEBLATE_ADDONS

2.16.25 DEFAULT_COMMITER_EMAIL

Novo na versão 2.4.

Committer e-mail address defaulting to noreply@weblate.org.

Ver também:

DEFAULT_COMMITER_NAME

2.16.26 DEFAULT_COMMITER_NAME

Novo na versão 2.4.

Committer name defaulting to Weblate.

Ver 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.

Ver também:

Definições de idioma, Idioma fonte

2.16.28 DEFAULT_MERGE_STYLE

Novo na versão 3.4.

Mescla o estilo para quaisquer novos componentes.

- *rebase* - padrão
- *merge*

Ver também:

Configuração de componente, Estilo de mesclagem

2.16.29 DEFAULT_SHARED_TM

Novo na versão 3.2.

Configures default value of *Usar memória de tradução compartilhada* and *Contribuir com memória de tradução compartilhada*.

2.16.30 DEFAULT_TRANSLATION_PROPAGATION

Novo na versão 2.5.

Configuração padrão para propagação de tradução, sendo o padrão `True`.

Ver também:

Configuração de componente, Permitir propagação de tradução

2.16.31 DEFAULT_PULL_MESSAGE

Título para novas pull requests, sendo o padrão `'Update from Weblate'`.

2.16.32 ENABLE_AVATARS

Se deve-se ativar avatares baseados em Gravatar para os usuários. Por padrão, isto está ativado.

Avatares são buscados e armazenados em cache no servidor, diminuindo o risco de vazamento de informações privadas, acelerando a experiência do usuário.

Ver também:

Cache de avatares, AVATAR_URL_PREFIX, Avatars

2.16.33 ENABLE_HOOKS

Se deve-se habilitar ganchos remotos anônimos.

Ver também:

Ganchos de notificação

2.16.34 ENABLE_HTTPS

Se deve-se enviar links para Weblate como HTTPS ou HTTP. Esta configuração afeta os e-mails enviados e as URLs absolutas geradas.

In the default configuration this is also used for several Django settings related to HTTPS - it enables secure cookies, toggles HSTS or enables redirection to HTTPS URL.

The HTTPS redirection might be problematic in some cases and you might hit issue with infinite redirection in case you are using a reverse proxy doing SSL termination which does not correctly pass protocol headers to Django. Please tweak your reverse proxy configuration to emit X-Forwarded-Proto or Forwarded headers or configure `SECURE_PROXY_SSL_HEADER` to let Django correctly detect the SSL status.

Ver 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 usuários 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 pode ser encontrado.

2.16.37 GITLAB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores GitLab.

Dica: Use isso no caso de você 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 usuário GitLab usado para enviar merge requests para atualizações de tradução.

Ver também:

GITLAB_CREDENTIALS, *GitLab*

2.16.39 GITLAB_TOKEN

Novo na versão 4.3.

O token de acesso pessoal GitLab usado para fazer chamadas de API para atualizações de tradução.

Ver também:

GITLAB_CREDENTIALS, *GitLab*, *GitLab: Personal access token*

2.16.40 GITHUB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores GitHub.

Dica: Use isso no caso de você 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 usuário GitHub usado para enviar pull request para atualizações de tradução.

Ver também:

GITHUB_CREDENTIALS, *GitHub*

2.16.42 GITHUB_TOKEN

Novo na versão 4.3.

O token de acesso pessoal GitHub usado para fazer chamadas de API para enviar pull requests para atualizações de tradução.

Ver também:

GITHUB_CREDENTIALS, *GitHub*, *Creating a personal access token*

2.16.43 GOOGLE_ANALYTICS_ID

Google Analytics ID para ativar o monitoramento do Weblate usando o Google Analytics.

2.16.44 HIDE_REPO_CREDENTIALS

Ocultar credenciais de repositório da interface web. No caso de você ter URL do repositório com usuário e senha, o Weblate irá ocultá-la quando as informações relacionadas são mostradas aos usuários.

Por exemplo, em vez de `https://usuário:senha@git.example.com/repo.git`, ela vai mostrar apenas `https://git.example.com/repo.git`. Ela tenta limpar mensagens de erro VCS também de forma semelhante.

Nota: Isso está ativado por padrão.

2.16.45 HIDE_VERSION

Novo na versão 4.3.1.

Esconde informação de versão de usuários não autenticados. Isso também faz todos os links de documentação apontar para a última versão, ao invés da versão de combinação da documentação instalada atualmente.

Ocultar versão é uma prática de segurança recomendada em algumas empresas, mas não prevê invasores de descobri a versão sondando o comportamento.

Nota: Isso está desligado por padrão.

2.16.46 IP_BEHIND_REVERSE_PROXY

Novo na versão 2.14.

Indica se o Weblate está sendo usado atrás de um proxy reverso.

Se definido como `True`, o Weblate obtém endereço IP de um cabeçalho definido por `IP_PROXY_HEADER`.

Aviso: Certifique-se de que você está realmente usando um proxy reverso e que ele define este cabeçalho, caso contrário, os usuários poderão falsificar o endereço IP.

Nota: Isso não está ativado por padrão.

Ver também:

Executando 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.

Padrão é `HTTP_X_FORWARDED_FOR`.

Ver também:

Executando 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 você pode configurar qual endereço do cabeçalho é usado como endereço IP do cliente aqui.

Aviso: Configurar isso afeta a segurança da sua instalação, então você só deve configurá-la para usar proxies confiáveis para determinar o endereço IP.

O padrão é 0.

Ver também:

Executando 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 sua instância de Weblate mostra seus documentos legais.

Dica: Útil se você hospeda 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 de 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: Configurando este para valor em branco agora desabilita o alerta de licença.

Filtrar licenças da lista para mostrar. Isto também desabilita 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"}
```

A seguir, desativa o alerta de licença:

```
LICENSE_FILTER = set()
```

Ver também:

alerts

2.16.52 LICENSE_REQUIRED

Define se o atributo de licença em *Configuração de componente* é necessário.

Nota: Isso está desativado por padrão.

2.16.53 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Se o comprimento de uma determinada tradução deve ser limitado. A restrição é o comprimento do texto fonte * 10 caracteres.

Dica: Defina isso como `False` para permitir traduções mais longas (até 10.000 caracteres) independentemente do comprimento do texto fonte.

Nota: O padrão 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 arquivos gerados que serão servidos em `LOCALIZE_CDN_URL`.

Dica: No Hosted Weblate, é usada com `https://weblate-cdn.com/`.

Ver também:

CDN de localização JavaScript

2.16.55 LOGIN_REQUIRED_URLS

Uma lista de URLs para as quais você deseja exigir autenticação. (Além das regras padrão incorporadas ao Weblate).

Dica: Isso permite que você proteja com senha toda uma instalação usando:

```
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.

Ver também:

REQUIRE_LOGIN

2.16.56 LOGIN_REQUIRED_URLS_EXCEPTIONS

Lista de exceções para `LOGIN_REQUIRED_URLS`. Se não especificado, os usuários podem acessar a página de autenticação.

Algumas das exceções que você pode querer 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 você quer rastrear.

Nota: Essa integração não tem suporte ao Matomo Tag Manager.

Ver também:

MATOMO_URL

2.16.58 MATOMO_URL

URL completa (incluindo barra ao final) de uma instalação Matomo (anteriormente Piwik) que você deseja usar para rastrear o uso do Weblate. Por favor, consulte <<https://matomo.org/>> para mais detalhes.

Dica: Essa integração não tem suporte ao Matomo Tag Manager.

Por exemplo:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Ver 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 habilitados para uso.

Nota: Muitos dos serviços precisam de configuração adicional, como chaves de API, consulte sua documentação *Tradução de máquina* para mais detalhes.

```
MT_SERVICES = (
    "weblate.machinery.apertium.ApertiumAPYTranslation",
    "weblate.machinery.deepl.DeepLTranslation",
    "weblate.machinery.glosbe.GlosbeTranslation",
    "weblate.machinery.google.GoogleTranslation",
    "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    "weblate.machinery.mymemory.MyMemoryTranslation",
    "weblate.machinery.tmserver.AmagamaTranslation",
    "weblate.machinery.tmserver.TMServerTranslation",
    "weblate.machinery.yandex.YandexTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.machinery.saptranslationhub.SAPTranslationHub",
    "weblate.memory.machine.WeblateMemory",
)
```

Ver também:

Tradução de máquina, Sugestões automáticas

2.16.60 MT_APERTIUM_APY

URL do servidor Apertium-APy, <https://wiki.apertium.org/wiki/Apertium-apy>

Ver também:

Apertium, Tradução de máquina, Sugestões automáticas

2.16.61 MT_AWS_ACCESS_KEY_ID

ID da chave de acesso para Amazon Translate.

Ver também:

AWS, Tradução de máquina, Sugestões automáticas

2.16.62 MT_AWS_SECRET_ACCESS_KEY

Chave secreta da API para o Amazon Translate.

Ver também:

AWS, Tradução de máquina, Sugestões automáticas

2.16.63 MT_AWS_REGION

Nome da região para usar no Amazon Translate.

Ver também:

AWS, Tradução de máquina, Sugestões automáticas

2.16.64 MT_Baidu_ID

ID do cliente para a API do Baidu Zhiyun, você pode se registrar em <https://api.fanyi.baidu.com/api/trans/product/index>

Ver também:

Baidu API machine translation, Tradução de máquina, Sugestões automáticas

2.16.65 MT_Baidu_SECRET

Segredo do cliente para a API do Baidu Zhiyun, você pode se registrar em <https://api.fanyi.baidu.com/api/trans/product/index>

Ver também:

Baidu API machine translation, Tradução de máquina, Sugestões automáticas

2.16.66 MT_DEEPL_API_VERSION

Novo na versão 4.1.1.

Versão da API para usar com o serviço DeepL. A versão limita o escopo de uso:

v1 Destina-se a ferramentas CAT e é utilizável com assinatura baseada no usuário.

v2 Destina-se ao uso da API e a assinatura é baseada em uso.

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, seu padrão é v2, e você pode alterar isso para v1 no caso de você ter uma assinatura CAT existente e querer que o Weblate use isso.

Ver também:

DeepL, Tradução de máquina, Sugestões automáticas

2.16.67 MT_DEEPL_KEY

Chave de API para a API do DeepL, você pode se registrar em <https://www.deepl.com/pro.html>

Ver também:

DeepL, Tradução de máquina, Sugestões automáticas

2.16.68 MT_GOOGLE_KEY

Chave de API para a API v2 do Google Translate, você pode se registrar em <https://cloud.google.com/translate/docs>

Ver também:

Google Translate, Tradução de máquina, Sugestões automáticas

2.16.69 MT_GOOGLE_CREDENTIALS

Arquivo 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 a determinado projeto. Por favor, verifique <https://cloud.google.com/docs/authentication/getting-started> para mais detalhes.

2.16.70 MT_GOOGLE_PROJECT

ID de projeto da API v3 do Google Cloud com serviço de tradução ativado e cobrança ativado. Por favor, consulte <https://cloud.google.com/appengine/docs/standard/nodejs/building-app/creating-project> para mais detalhes

2.16.71 MT_GOOGLE_LOCATION

A API v3 do App Engine do Google Cloud pode ser específica para um local. Altere conforme o caso, se o padrão `global` não servir para você.

Consulte <https://cloud.google.com/appengine/docs/locations> para mais detalhes

Ver também:

Google Translate API V3 (Advanced)

2.16.72 MT_MICROSOFT_BASE_URL

Domínio de URL base da região conforme definido na seção “URLs base”.

O padrão é `api.cognitive.microsofttranslator.com` para o Azure Global.

Para Azure China, use `api.translator.azure.cn`.

2.16.73 MT_MICROSOFT_COGNITIVE_KEY

Chave do cliente para a API do Microsoft Cognitive Services Translator.

Ver também:

Microsoft Cognitive Services Translator, Tradução de máquina, Sugestões automáticas, Serviços Cognitivos - API de Tradução de Texto, Microsoft Azure Portal

2.16.74 MT_MICROSOFT_REGION

Prefixo da região conforme definido na seção “Authenticating with a Multi-service resource” <<https://docs.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-reference#authenticating-with-a-multi-service-resource>>.

2.16.75 MT_MICROSOFT_ENDPOINT_URL

Domínio de URL de extremidade da região para token de acesso definido na seção “Autenticando com um token de acesso”.

O padrão é `api.cognitive.microsoft.com` para Azure Global.

Para Azure China, use sua extremidade do Portal do Azure.

2.16.76 MT_MODERNMT_KEY

Chave de API para o mecanismo de tradução de máquina ModernMT.

Ver também:

ModernMT MT_MODERNMT_URL

2.16.77 MT_MODERNMT_URL

URL de ModernMT. Seu padrão é `https://api.modernmt.com/` para o serviço de nuvem.

Ver também:

ModernMT MT_MODERNMT_KEY

2.16.78 MT_MYMEMORY_EMAIL

Endereço de e-mail de identificação do myMemory. Permite 1000 solicitações por dia.

Ver também:

MyMemory, *Tradução de máquina*, *Sugestões automáticas*, *MyMemory: API technical specifications*

2.16.79 MT_MYMEMORY_KEY

Chave de acesso do MyMemory para memória de tradução privada. Use-a com *MT_MYMEMORY_USER*.

Ver também:

MyMemory, *Tradução de máquina*, *Sugestões automáticas*, *MyMemory: API key generator*

2.16.80 MT_MYMEMORY_USER

ID de usuário do MyMemory para memória de tradução privada. Use-o com *MT_MYMEMORY_KEY*.

Ver também:

MyMemory, *Tradução de máquina*, *Sugestões automáticas*, *MyMemory: API key generator*

2.16.81 MT_NETEASE_KEY

Chave de aplicativo para API da NetEase Sight, você pode se registrar em <https://sight.youdao.com/>

Ver também:

NetEase Sight API machine translation, *Tradução de máquina*, *Sugestões automáticas*

2.16.82 MT_NETEASE_SECRET

Segredo de aplicativo para a API da NetEase Sight, você pode se registrar em <https://sight.youdao.com/>

Ver também:

NetEase Sight API machine translation, *Tradução de máquina*, *Sugestões automáticas*

2.16.83 MT_TMSERVER

URL onde o tmserver está funcionando.

Ver também:

tmserver, *Tradução de máquina*, *Sugestões automáticas*, *tmserver*

2.16.84 MT_YANDEX_KEY

Chave de API para a API do Yandex Translate, você pode se registrar em <https://yandex.com/dev/translate/>

Ver também:

Yandex Translate, *Tradução de máquina*, *Sugestões automáticas*

2.16.85 MT_YOUDAO_ID

ID do cliente para a API do Youdao Zhiyun, você pode se registrar em <https://ai.youdao.com/product-fanyi-text.s>.

Ver também:

Youdao Zhiyun API machine translation, Tradução de máquina, Sugestões automáticas

2.16.86 MT_YOUDAO_SECRET

Segredo do cliente para a API do Youdao Zhiyun, você pode se registrar em <https://ai.youdao.com/product-fanyi-text.s>.

Ver também:

Youdao Zhiyun API machine translation, Tradução de máquina, Sugestões automáticas

2.16.87 MT_SAP_BASE_URL

URL de API para o serviço SAP Translation Hub.

Ver também:

SAP Translation Hub, Tradução de máquina, Sugestões automáticas

2.16.88 MT_SAP_SANDBOX_APIKEY

Chave de API para uso de API em caixa de proteção

Ver também:

SAP Translation Hub, Tradução de máquina, Sugestões automáticas

2.16.89 MT_SAP_USERNAME

Seu nome de usuário SAP

Ver também:

SAP Translation Hub, Tradução de máquina, Sugestões automáticas

2.16.90 MT_SAP_PASSWORD

Sua senha SAP

Ver também:

SAP Translation Hub, Tradução de máquina, Sugestões automáticas

2.16.91 MT_SAP_USE_MT

Se deve também usar serviços de tradução de máquina, além do banco de dados de termos. Possíveis valores: `True` ou `False`

Ver também:

SAP Translation Hub, Tradução de máquina, Sugestões automáticas

2.16.92 NEARBY_MESSAGES

Quantos textos devem ser mostrados em torno do texto traduzido atualmente. Este é apenas um valor padrão, os usuários podem ajustar isso em *Perfil do usuário*.

2.16.93 PAGURE_CREDENTIALS

Novo na versão 4.3.2.

Lista para credenciais para servidores Pagure.

Dica: Use isso no caso de você 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",
    },
}
```

2.16.94 PAGURE_USERNAME

Novo na versão 4.3.2.

O nome de usuário Pagure usado para enviar merge requests para atualizações de tradução.

Ver também:

PAGURE_CREDENTIALS, Pagure

2.16.95 PAGURE_TOKEN

Novo na versão 4.3.2.

O token de acesso pessoal Pagure usado para fazer chamadas de API para atualizações de tradução.

Ver também:

PAGURE_CREDENTIALS, Pagure, Pagure API

2.16.96 RATELIMIT_ATTEMPTS

Novo na versão 3.2.

O número máximo de tentativas de autenticação antes da limitação da taxa ser aplicada.

O padrão é 5.

Ver também:

Limitação de taxa, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

2.16.97 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 tendo como padrão 300 (5 minutos).

Ver também:

Limitação de taxa, `RATELIMIT_ATTEMPTS`, `RATELIMIT_LOCKOUT`

2.16.98 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 tendo como padrão 600 (10 minutos).

Ver também:

Limitação de taxa, `RATELIMIT_ATTEMPTS`, `RATELIMIT_WINDOW`

2.16.99 REGISTRATION_ALLOW_BACKENDS

Novo na versão 4.1.

A lista de backends de autenticação de onde permite o registro. Isso só limita novos registros, os usuários ainda podem autenticar e adicionar autenticação usando todos os backends de autenticação configurados.

É recomendado para manter `REGISTRATION_OPEN` habilitado enquanto limita os backends de registro, caso contrário, os usuários poderão se registrar, mas o Weblate não mostrará links para se registrar na interface do usuário.

Exemplo:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Dica: Os nomes de backend correspondem aos nomes usados na URL para autenticação.

Ver também:

`REGISTRATION_OPEN`, *Autenticação*

2.16.100 REGISTRATION_CAPTCHA

Um valor de `True` ou `False` indicando se o registro de novas contas é protegido pelo CAPTCHA. Esta configuração é opcional, e um padrão de `True` será presumido se não for fornecido.

Se ativado, um CAPTCHA é adicionado a todas as páginas onde um usuário digita seu endereço de e-mail:

- Registro de nova conta.
- Recuperação de senha.
- Adição de e-mail a uma conta.
- Formulário de contato para usuários que não estão autenticados.

2.16.101 REGISTRATION_EMAIL_MATCH

Novo na versão 2.17.

Permite filtrar quais endereços de e-mail podem ser registrados.

O padrão é `.*`, que permite que qualquer endereço de e-mail seja registrado.

Você pode usá-lo para restringir o registro a um único domínio de e-mail:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.102 REGISTRATION_OPEN

Se o registro de novas contas é atualmente permitido. Essa configuração opcional pode permanecer com o padrão `True`, ou pode ser alterada para `Falsa`.

Essa configuração afeta a autenticação embutida por endereço de e-mail ou através do Python Social Auth (você pode listar certos back-ends usando [REGISTRATION_ALLOW_BACKENDS](#)).

Nota: Se estiver usando métodos de autenticação de terceiros, como [Autenticação por LDAP](#), ele apenas oculta o formulário de registro, mas novos usuários ainda conseguem se autenticar e criar contas.

Ver também:

[REGISTRATION_ALLOW_BACKENDS](#), [REGISTRATION_EMAIL_MATCH](#), [Autenticação](#)

2.16.103 REPOSITORY_ALERT_THRESHOLD

Novo na versão 4.0.2.

Limiar para acionar um alerta para repositórios desatualizados ou aqueles que contenham muitas alterações. O padrão é 25.

Ver também:

[alerts](#)

2.16.104 REQUER_LOGIN

Novo na versão 4.1.

This enables `LOGIN_REQUIRED_URLS` and configures REST framework to require authentication for all API endpoints.

Nota: Isto é implementado no *Amostra de configuração*. Para Docker, use `WEBLATE_REQUIRE_LOGIN`.

2.16.105 SENTRY_DSN

Novo na versão 3.9.

DSN do Sentry para usar para *Coletando relatórios de erros*.

Ver também:

[Integração Django para o Sentry](#)

2.16.106 IDADE_REGISTRO_SESSÃO_AUTENTICADO

Novo na versão 4.3.

Configura a expiração da sessão para usuário autenticados. Isso complementa `IDADE_REGISTRO_SESSÃO` que é utilizado por usuários não autenticados.

Ver também:

`SESSION_COOKIE_AGE`

2.16.107 SIMPLIFY_LANGUAGES

Usa códigos de idioma simples para combinações padrão 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 padrão.

Desative isso se quiser traduções diferentes para cada variante.

2.16.108 SITE_DOMAIN

Configura o domínio do site. Isso é necessário para produzir links absolutos corretos em muitos escopos (por exemplo, ativação de e-mails, notificações ou feeds RSS).

No caso de o Weblate estar sendo executado em um 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 (habilitar e aplicar HTTPS), use `ENABLE_HTTPS` e for alterar URL, use `URL_PREFIX`.

Dica: Em um contêiner Docker, o domínio do site é configurado através de `WEBLATE_ALLOWED_HOSTS`.

Ver também:

Definir domínio correto do site, Configuração de hosts permitidos, Configurar corretamente HTTPS WE-
BLATE_SITE_DOMAIN, ENABLE_HTTPS

2.16.109 SITE_TITLE

Título do site a ser usado para o site e e-mails enviados.

2.16.110 SPECIAL_CHARS

Caracteres adicionais para incluir no teclado visual, *Teclado visual*.

O valor padrão é:

```
SPECIAL_CHARS = ("\\t", "\\n", "...")
```

2.16.111 SINGLE_PROJECT

Novo na versão 3.8.

Redireciona os usuários diretamente para um projeto ou componente em vez de mostrar o painel. Você pode configurá-lo como `True` e, neste caso, ele só funciona no caso de haver realmente apenas um único projeto no Weblate. Alternativamente, defina o projeto, e ele 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"
```

2.16.112 STATUS_URL

A URL onde sua instância de Weblate relata seu status.

2.16.113 SUGGESTION_CLEANUP_DAYS

Novo na versão 3.2.1.

Exclui automaticamente sugestões após um determinado número de dias. O padrão é `None`, ou seja, sem exclusões.

2.16.114 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`.

Ver também:

Built-in language definitions

2.16.115 URL_PREFIX

Esta configuração permite que você execute Weblate em algum caminho (caso contrário, ele depende de ser executado a partir da raiz do servidor web).

Nota: Para usar esta configuração, você também precisa configurar 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, você teria que ajustar `urls.py` para conter este prefixo.

2.16.116 VCS_BACKENDS

Configuração de backends VCS disponíveis.

Nota: Weblate tenta usar todos os back-ends suportados para os seus usuários.

Dica: Você pode limitar escolhas ou adicionar back-ends VCS personalizados usando isso.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Ver também:

Integração com controle de versão

2.16.117 VCS_CLONE_DEPTH

Novo na versão 3.10.2.

Configura o quão profunda a clonagem de repositórios Weblate deve ir.

Nota: Atualmente, isso só é suportado em [Git](#). Por padrão, o Weblate faz clones rasos dos repositórios para tornar a clonagem mais rápida e economizar espaço em disco. Dependendo do seu uso (por exemplo, ao usar o personalizado [Extensões](#)), você pode querer aumentar a profundidade ou desligar os clones rasos completamente definindo isso para 0.

Dica: No caso de você receber 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
```

2.16.118 WEBLATE_ADDONS

Lista de extensões disponíveis para uso. Para usá-las, elas devem ser habilitadas para um determinado componente de tradução. Por padrão, isso inclui todas as extensões embutidas, ao estender a lista, você provavelmente vai querer manter as existentes habilitadas, 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",
    # Addon 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.

Ver também:*Extensões, DEFAULT_ADDONS*

2.16.119 WEBLATE_EXPORTERS

Novo na versão 4.2.

Lista de exportadores disponíveis que oferecem download de traduções ou glossários em vários formatos de arquivo.

Ver também:*Formatos de arquivos suportados*

2.16.120 WEBLATE_FORMATS

Novo na versão 3.0.

Lista de formatos de arquivo disponíveis para uso.

Nota: A lista padrão já tem os formatos comuns.

Ver também:*Formatos de arquivos suportados*

2.16.121 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.

Ver também:*Signing Git commits with GnuPG*

2.16.122 WEBSITE_REQUIRED

Defines whether *Site do projeto* has to be specified when creating a project. Turned on by default as that suits public server setups.

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
```

(continua na próxima página)

(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

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # Uncomment for MySQL older than 5.7:
            # "init_command": "SET sql_mode='STRICT_TRANS_TABLES'",
            # Set emoji capable charset for MySQL:
            # "charset": "utf8mb4",
            # Change connection timeout in case you get MySQL gone away error:
            # "connect_timeout": 28800,
        },
    },
}
```

(continua na próxima página)

(continuação da página anterior)

```

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", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "简体中文"),
    ("zh-hant", "繁體中文"),

```

(continua na próxima página)

```

)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not format dates, numbers and
# calendars according to the current locale.
USE_L10N = True

# If you set this to False, Django will not use timezone-aware datetimes.
USE_TZ = True

# Type of automatic primary key, introduced in Django 3.2
DEFAULT_AUTO_FIELD = "django.db.models.AutoField"

# URL prefix to use, please see documentation for more details
URL_PREFIX = ""

# Absolute filesystem path to the directory that will hold user-uploaded files.
MEDIA_ROOT = os.path.join(DATA_DIR, "media")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
MEDIA_URL = f"{URL_PREFIX}/media/"

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
STATIC_ROOT = os.path.join(DATA_DIR, "static")

# URL prefix for static files.
STATIC_URL = f"{URL_PREFIX}/static/"

# Additional locations of static files
STATICFILES_DIRS = (
    # Put strings here, like "/home/html/static" or "C:/www/django/static".
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
)

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    "django.contrib.staticfiles.finders.FileSystemFinder",
    "django.contrib.staticfiles.finders.AppDirectoriesFinder",
    "compressor.finders.CompressorFinder",
)

# Make this unique, and don't share it with anybody.
# 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)]

```

(continua na próxima página)

(continuação da página anterior)

```

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
            "loaders": _TEMPLATE_LOADERS,
        },
    },
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_BITBUCKET_KEY = ""
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 = ""

```

(continua na próxima página)


```

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "weblate.accounts.pipeline.password_reset",
)

SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/#account
↪"
SOCIAL_AUTH_PROTECTED_USER_FIELDS = ("email",)
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
    },
    {

```

(continua na próxima página)

(continuação da página anterior)

```

        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"},
    # Optional password strength validation by django-zxcvbn-password
    # {
    #     "NAME": "zxcvbn_password.ZXCVBNValidator",
    #     "OPTIONS": {
    #         "min_score": 3,
    #         "user_attributes": ("username", "email", "full_name")
    #     }
    # },
]

# Password hashing (prefer Argon)
PASSWORD_HASHERS = [
    "django.contrib.auth.hashers.Argon2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher",
    "django.contrib.auth.hashers.BCryptSHA256PasswordHasher",
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # 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",

```

(continua na próxima página)

(continuação da página anterior)

```

"weblate.memory",
"weblate.screenshots",
"weblate.fonts",
"weblate.accounts",
"weblate.configuration",
"weblate.utils",
"weblate.vcs",
"weblate.wladmin",
"weblate.metrics",
"weblate",
# Optional: Git exporter
"weblate.gitexport",
# Standard Django modules
"django.contrib.auth",
"django.contrib.contenttypes",
"django.contrib.sessions",
"django.contrib.messages",
"django.contrib.staticfiles",
"django.contrib.admin.apps.SimpleAdminConfig",
"django.contrib.admindocs",
"django.contrib.sitemaps",
"django.contrib.humanize",
# Third party Django modules
"social_django",
"crispy_forms",
"compressor",
"rest_framework",
"rest_framework.authtoken",
"django_filters",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↳WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - you can also choose "logfile" to log into separate file
#   after configuring it below

# Detect if we can connect to syslog
HAVE_SYSLOG = False
if platform.system() != "Windows":
    try:
        handler = SysLogHandler(address="/dev/log", facility=SysLogHandler.LOG_
↳LOCAL2)
        handler.close()
        HAVE_SYSLOG = True
    except OSError:
        HAVE_SYSLOG = False

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.

```

(continua na próxima página)

(continuação da página anterior)

```

# 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",
        # },
    },
    "loggers": {
        "django.request": {
            "handlers": ["mail_admins", DEFAULT_LOG],
            "level": "ERROR",
            "propagate": True,
        },
        "django.server": {
            "handlers": ["django.server"],
            "level": "INFO",
            "propagate": False,
        },
    },
}

```

(continua na próxima página)

```

    },
    # Logging database queries
    # "django.db.backends": {
    #     "handlers": [DEFAULT_LOG],
    #     "level": "DEBUG",
    # },
    "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Logging VCS operations
    "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Python Social Auth
    "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Django Authentication Using LDAP
    "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # SAML IdP
    "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    #     "weblate.machinery.apertium.ApertiumAPYTranslation",
    #     "weblate.machinery.baidu.BaiduTranslation",
    #     "weblate.machinery.deepl.DeepLTranslation",
    #     "weblate.machinery.glosbe.GlosbeTranslation",
    #     "weblate.machinery.google.GoogleTranslation",
    #     "weblate.machinery.googlev3.GoogleV3Translation",
    #     "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    #     "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    #     "weblate.machinery.modernmt.ModernMTTranslation",
    #     "weblate.machinery.mymemory.MyMemoryTranslation",
    #     "weblate.machinery.netease.NeteaseSightTranslation",
    #     "weblate.machinery.tmserver.AmagamaTranslation",
    #     "weblate.machinery.tmserver.TMServerTranslation",
    #     "weblate.machinery.yandex.YandexTranslation",
    #     "weblate.machinery.saptranslationhub.SAPTranslationHub",
    #     "weblate.machinery.youdao.YoudaoTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APy server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# 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

```

(continuação da página anterior)

```

# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_Baidu_ID = None
MT_Baidu_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.
# ↪html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS

# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view

```

(continua na próxima página)

(continuação da página anterior)

```

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 referrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",) # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

```

(continua na próxima página)

(continuação da página anterior)

```

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PerlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.SchemeFormatCheck",
#     "weblate.checks.format.CSharpFormatCheck",
#     "weblate.checks.format.JavaFormatCheck",
#     "weblate.checks.format.JavaMessageFormatCheck",
#     "weblate.checks.format.PercentPlaceholdersCheck",
#     "weblate.checks.format.VueFormattingCheck",
#     "weblate.checks.format.I18NextInterpolationCheck",
#     "weblate.checks.format.ESTemplateLiteralsCheck",
#     "weblate.checks.angularjs.AngularJSInterpolationCheck",
#     "weblate.checks.qt.QtFormatCheck",
#     "weblate.checks.qt.QtPluralCheck",
#     "weblate.checks.ruby.RubyFormatCheck",
#     "weblate.checks.consistency.PluralsCheck",
#     "weblate.checks.consistency.SamePluralsCheck",
#     "weblate.checks.consistency.ConsistencyCheck",
#     "weblate.checks.consistency.TranslatedCheck",
#     "weblate.checks.chars.EscapedNewlineCountingCheck",
#     "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",

```

(continua na próxima página)

(continuação da página anterior)

```

# "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"

# 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

```

(continua na próxima página)

(continuação da página anterior)

```

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",
    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL
FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = [

```

(continua na próxima página)

(continuação da página anterior)

```

    {"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"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

```

(continua na próxima página)

(continuação da página anterior)

```
# PGP commits signing
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None
SENTRY_DSN = None
SENTRY_ENVIRONMENT = None
AKISMET_API_KEY = None
```

2.18 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 gerenciamento

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
```

Ver também:

Instalando usando 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
```

Ver 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*.

Ver também:

Executando 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.

Ver também:

Executando 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.

Ver também:

Memória de tradução, Esquema de memória de tradução do 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"
  }
]
```

Ver 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.

Ver também:

Memória de tradução, Esquema de memória de tradução do 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 componente* addon, leading to some changes in behavior and what parameters are accepted.

Batch imports components into project based on filemask.

<project> names an existing project, into which the components are to be imported.

The <gitrepo> defines the Git repository URL to use, and <branch> signifies the Git branch. To import additional translation components from an existing Weblate component, use a `weblate://<project>/<component>` URL for the <gitrepo>.

The <filemask> defines file discovery for the repository. It can be either be made simple using wildcards, or it can use the full power of regular expressions.

The simple matching uses `**` for component name and `*` for language, for example: `**/*.po`

The regular expression has to contain groups named *component* and *language*. For example: `(?P<language>[^\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 arquivos 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'
```

Ver 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>

Installs an addon to a set of components.

--addon ADDON

Name of the addon to install. For example `weblate.gettext.customize`.

--configuration CONFIG

JSON encoded configuration of an addon.

--update

Update the existing addon configuration.

You can either define which project or component to install the addon in (for example `weblate/application`), or use `--all` to include all existing components.

To install *Personalizar saída do gettext* for all components:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --  
↪update --all
```

Ver 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.l10n.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.

Ver 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 *Push ao fazer commit* 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.

Ver 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 [*Controle de acesso do projeto*](#).

Ver 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 *Ganchos 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, status ou especificar metas para tradução.

Os usuários receberão notificação sobre os anúncios de projetos assistidos (a menos que optem por não participar).

Isso 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*, usando `: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 4.6 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Ele também pode ser adicionado usando a interface administrativa:

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.

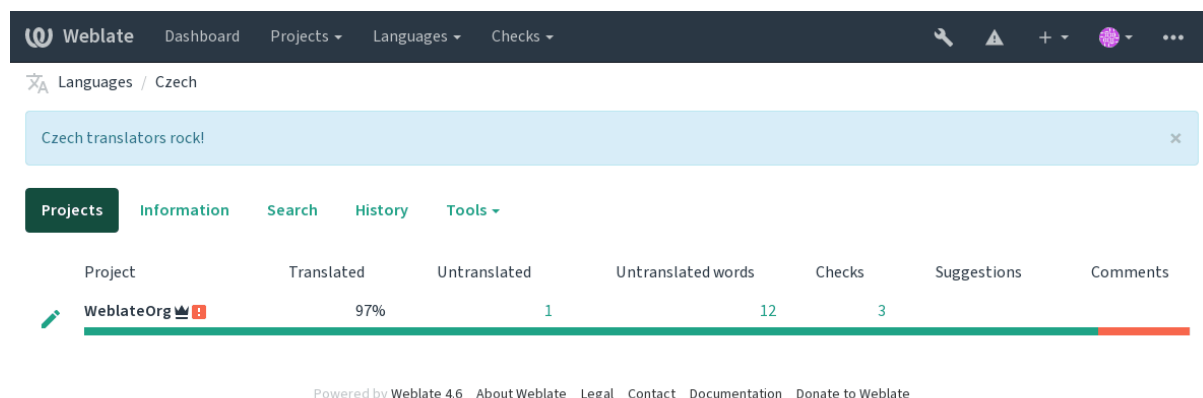
Componente especificado

Mostrado para um determinado componente e todas as suas traduções.

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 usuário, a partir do qual os usuários podem selecionar uma visualização como exibição padrão. Veja [Painel](#) para saber mais.

Alterado na versão 2.20: Um status vai ser apresentado para cada componente listado no painel.

Os nomes e conteúdos das listas de componentes podem ser especificados na interface administrativa, na seção *Component lists*. Cada lista de componentes deve ter um nome que é exibido para o usuário e uma slug representando-a na URL.

Alterado na versão 2.13: Altera as configurações de painel para usuários anônimos a partir da interface administrativa, alterando qual painel é apresentado para usuários não autenticados.

2.20.1 Listas de componentes automáticas

Novo na versão 2.13.

Adicione componentes à lista automaticamente com base em suas slug criando regras *Automatic component list assignment*.

- Útil para manutenção de listas de componentes para instalações de grandes dimensões, ou no caso de você querer uma lista de componentes com todos os componentes em sua instalação de 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

Save and add another

Save and continue editing

SAVE

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 Cobrança

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 usuário*.

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

Ver 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 usuário não autenticados
- Project and component descriptions and links

Nota: This (among other things) relies on IP address of the client, please see *Executando por trás de um proxy reverso* for properly configuring that.

Ver também:

Executando 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 D3386879A59B6F9BD780039D5E3D09F5D28D1902, for which the corresponding public key is found below.

```

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGB8E4sBDADi7OXDBHNKpYGL0qwUwY25HkntNcbWhtocalTddUiGuafo/cG7
njMwhfir98OMcfaXsc/tYmjii/phWtsK0cR4rYuZcQFERYyTcMRFAMZcUfqz8xx
DyNO05sZDAZtW/5Ra6n+Dg+FWJ0xPWZAsIjFs+V2NOIKdIAW05W4Tb4gp72qPQTE
Ts3iXoFb8uQNuYD6sxQegpZpJGlpe/nw/GzmqkvKAKJxx7XlyjgWaZgpANhbTXK
htKV/TyCP6XZbvRlvLaKWPPNvMgFHc8aWjZyOmr/b3al42deewxJNa0DJyDMsx
hCrC2lAs4ysk/hZFo8lQ6+rabwXjw9CRJAGm/YGFwsovv4godCT16icjYC7sJaC5
z0dL9chSmMJMsZRB+5nZTdlvfqzeN342LkojAlz2VEJnm8EHAU65kCfHl3v7Ml
SWuPgJTCnEa3gUHFILWHn9ohI3jmUP2b57ZQzAlmISZy+jsm9UbqootQit6ecqlg
5rk6lJgvLeX+NKcAEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWIQTTOGH5pZtvm9eAA51ePQn10o0ZAgUCYHx7iwlBawULCQgHAgYV
CgkICwIEFgIDAQIeAQIXgAAKCRBePQn10o0ZAnxkC/sGm1V2CwpVwOI73lzK8nli
JEhVHFrDeOwkXhN96hczUJqocZAmseUDgQ4eeY4YW55/QCGIHo0wz3XoslCp00k
c6qFluBIUWjXUyTs9vTcf3eU5tcPxxaaXIVozkdAx1sZlvB7H7ODvuUYIQDjXJnk
6iPh7z6VW4R6+eLNG5z+nOii3h4l7l7PY//EezQ4Sx7kdvAkKvT7S/nT7vx9IGHV

```

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

2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking `gpg`.

Ver 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	Escopo	Allowed attempts	Rate limit window	Lockout period
Registro	REGISTRATION	5	300	600
Sending message to admins	MESSAGE	5	300	600
Password authentication on sign in	LOGIN	5	300	600
Sitewide search	SEARCH	6	60	60
Traduzindo	TRANSLATE	30	60	600
Adding to glossary	GLOSSARY	30	60	600
Starting translation into a new language	LANGUAGE	2	300	600

If a user fails to log in `AUTH_LOCK_ATTEMPTS` times, password authentication will be turned off on the account until having gone through the process of having its password reset.

The settings can be also applied in the Docker container by adding `WEBLATE_` prefix to the setting name, for example `RATELIMIT_ATTEMPTS` becomes `WEBLATE_RATELIMIT_ATTEMPTS`.

The API has separate rate limiting settings, see *Limitação de taxa da API*.

Ver também:

Limitação de taxa, Executando por trás de um proxy reverso, Limitação de taxa da API

2.21.8 Fedora Messaging integration

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 Personalizando o Weblate

Amplie e personalize usando Django e Python. Contribua suas alterações para o upstream acima para que todos possam se beneficiar. Isso reduz 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 suas próprias personalizações para cada atualização, as interfaces ou sua semântica podem mudar sem aviso prévio.

Ver também:

Contribuindo para o Weblate

2.22.1 Criando um módulo Python

Se você não está familiarizado com Python, você pode querer olhar para [Python For Beginners](#), explicando o básico e apontando para os 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 sua personalização em um pacote Python adequado:

1. Crie uma pasta para o seu pacote (usaremos `weblate_customization`).
2. Dentro dele, crie um arquivo `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",
```

(continua na próxima página)

(continuação da página anterior)

```
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 arquivo `__init__.py` para garantir que o Python possa importar o módulo.
5. Este pacote agora pode ser instalado usando **`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
```

Você pode encontrar um exemplo de personalização do Weblate em <https://github.com/WeblateOrg/customize-example>, ele abrange todos os tópicos descritos abaixo.

2.22.2 Alterando o logotipo

1. Crie um aplicativo Django simples contendo os arquivos estáticos que deseja substituir (veja *Criando um módulo Python*).

A marca aparece nos seguintes arquivos:

`icons/weblate.svg` Logotipo mostrado na barra de navegação.

`logo-*.png` Ícones web dependendo da resolução da tela e do navegador web.

`favicon.ico` Ícone web usado por navegadores legados.

`weblate-*.png` Avatares para bots ou usuários anônimos. 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 arquivos estáticos servidos aos clientes.

Ver também:

Gerenciamento de arquivos estáticos (ex.: imagens, JavaScript, CSS), *Servindo arquivos estáticos*

2.22.3 Verificações de qualidade personalizadas, extensões e correções automáticas

To install your code for *Correções automáticas personalizadas*, *Escrevendo as próprias verificações* or *Escrevendo extensões* in Weblate:

1. Place the files into your Python module containing the Weblate customization (see *Criando um módulo Python*).
2. Adicione seu caminho totalmente qualificado 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",)

# Addons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Ver também:

autocorreção-personalizada, verificações-próprias, extensão-própria, script-extensão

2.23 Interface de gerenciamento

A interface de gerenciamento oferece configurações de administração sob a URL `/manage/`. Está disponível para usuários que se inscrevem com privilégios administrativos, acessíveis usando 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 a sidebar containing 'Weblate status', 'Backups', 'Translation memory', 'Performance report', 'SSH keys', 'Alerts', 'Repositories', 'Users', and 'Appearance'. The main content area shows the 'Weblate support status' section, which includes the 'Weblate version' (4.6), 'Support status' (Community support), and buttons for 'Purchase support package' and 'Donate to Weblate'. Below this is the 'Activate support package' section, which includes a text box for the 'Activation token' and buttons for 'Activate' and 'Purchase support package'. At the bottom, there's a footer with 'Powered by Weblate 4.6' and links for 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

It includes basic overview of your Weblate:

- Support status, see *Obtendo suporte para o Weblate*
- Backups, see *Fazendo backup e movendo o Weblate*

- Memória de tradução compartilhada, veja *Memória de tradução*
- Performance report to review Weblate health and length of Celery queues
- Gerenciamento 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 seu uso é desencorajado — a maioria das funcionalidades podem ser gerenciadas diretamente no Weblate.

Aqui você pode gerenciar objetos armazenados no banco de dados, tais como usuários, traduções e outras configurações:

Webplate administration

WELCOME WEPLATE TEST · RETURN TO WEPLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT

Site administration

REPORTS

Webplate support status

Status of repositories

SSH keys

Performance report

Translation memory

ACCOUNTS

Audit logs

+ Add

Change

Profiles

+ Add

Change

Verified emails

+ Add

Change

AUTH TOKEN

Tokens

+ Add

Change

AUTHENTICATION

Groups

+ Add

Change

Roles

+ Add

Change

Users

+ Add

Change

BILLING

Billings

+ Add

Change

Invoices

+ Add

Change

Plans

+ Add

Change

FONTS

Font groups

+ Add

Change

Fonts

+ Add

Change

LEGAL

Agreements

+ Add

Change

PYTHON SOCIAL AUTH

Associations

+ Add

Change

Nonces

+ Add

Change

User social auths

+ Add

Change

SCREENSHOTS

Screenshots

+ Add

Change

TRANSLATION MEMORY

Memorys

+ Add

Change

WEPLATE CONFIGURATION

Settings

+ Add

Change

WEPLATE LANGUAGES

Languages

+ Add

Change

WEPLATE 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 seção *Relatórios*, você pode verificar o status de seu site, ajustá-lo para produção ou gerenciar chaves SSH usadas para acessar *Acessando repositórios*.

Gerencie objetos de banco de dados sob qualquer uma das seções. A mais interessante é provavelmente *Traduções do Weblate*, onde você pode gerenciar projetos traduzíveis, veja *Configuração de projeto* e *Configuração de componente*.

Idiomas do Weblate detém as definições de idiomas, explicado melhor em *Definições de idioma*.

Adicionando um projeto

A adição de um projeto serve como contêiner para todos os componentes. Normalmente você cria um projeto para um software, ou livro (Veja *Configuração de projeto* para informações sobre parâmetros individuais):

The screenshot shows the 'Add Project' form in the Weblate administration interface. The header includes 'Weblate administration' and a navigation bar with links: 'WELCOME, WEBLATE TEST, RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT'. The breadcrumb trail is 'Home > Weblate translations > Projects > Add Project'.

The form is titled 'Add Project' and includes a note: 'Required fields are marked in bold.' The fields are as follows:

- Project name:** A text input field containing 'WeblateOrg'. Below it, a smaller text says 'Display name'.
- URL slug:** A text input field containing 'weblateorg'. Below it, a smaller text says 'Name used in URLs and filenames.'
- Project website:** A text input field containing 'https://weblate.org/'. Below it, a smaller text says 'Main website of translated project.'
- Translation instructions:** A large text area containing 'https://weblate.org/contribute/'. Below it, a smaller text says 'You can use Markdown and mention users by @username.'
- Checkboxes:**
 - ☒ **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:** A dropdown menu set to 'Protected'. Below it, a smaller text says '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:** A text input field. Below it, a smaller text says 'Comma-separated list of language code mappings, for example: en_GB:en,en_US:en'.

At the bottom right, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Ver também:

Configuração de projeto

Componentes bilíngues

Uma vez adicionado um projeto, os componentes de tradução podem ser adicionados a ele. (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

[illegible]

Ver também:

Configuração de componente, Formatos bilíngues e monolíngues

Componentes monolíngues

Para facilitar a tradução destes, forneça um arquivo de modelo contendo o mapeamento de IDs de mensagem para seu respectivo idioma fonte (geralmente inglês). (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

[illegible]

Ver também:*Configuração de componente, Formatos bilíngues e monolíngues*

2.24 Obtendo suporte para o Weblate

Weblate é um software livre protegido por copyleft e com suporte comunitário. Os assinantes recebem suporte prioritário sem custo adicional. Pacotes de ajuda pré-pago estão disponíveis para todos. Você pode encontrar mais informações sobre as ofertas de suporte atuais em <<https://weblate.org/support/>>.

2.24.1 Integrando suporte

Novo na versão 3.8.

Os pacotes de suporte adquiridos podem ser integrados opcionalmente ao seu [gerenciamento de assinatura](#) do Weblate, de onde você encontrará um link para ele. Detalhes básicos da instância sobre sua instalação também são relatados de volta ao Weblate desta forma.

The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo and links for Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with a grid of buttons: Weblate status (highlighted), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, and Appearance. Underneath are 'Tools' and 'Billing' links. The main content area has two panels. The first panel, 'Weblate support status', shows the Weblate version as '4.6' with a long alphanumeric hash, the support status as 'Community support', and two buttons: 'Purchase support package' and 'Donate to Weblate'. The second panel, 'Activate support package', contains a text box for an 'Activation token' and a note about the support packages including priority email support or cloud backups. It also has 'Activate' and 'Purchase support package' buttons. At the bottom of the page, a footer line reads 'Powered by Weblate 4.6' followed by links for About Weblate, Legal, Contact, Documentation, and Donate to Weblate.

2.24.2 Dados enviados para a Weblate

- URL onde sua instância do Weblate está configurada
- Título do seu site
- A versão do Weblate que você está executando
- Contagem de alguns objetos em seu banco de dados Weblate (projetos, componentes, idiomas, textos fonte e usuários)
- 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 suporte ainda é válido
- *Armazenamento de backup provisionado do Weblate*
- *Descubra Weblate*

Dica: Os pacotes de suporte 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 em um beta inicial.

Discover Weblate is an opt-in service that makes it easier for users to find Weblate servers and communities. Users can browse registered services on <<https://weblate.org/discover/>>, and find there projects to contribute.

Getting listed

Dica: Participating in Discover Weblate makes Weblate submit some information about your server, please see *Dados enviados para a Weblate*.

To list your server with an active support subscription (see *Integrando suporte*) in Discover Weblate all you need to do is turn this on in the management panel:

Weblate
 Dashboard Projects Languages Checks

Manage

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

Tools
 Billing

Weblate support status ⓘ

Weblate version	4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c
Support status	Community support
Discover Weblate	Your Weblate is not listed on weblate.org Browse discovery
<div>Enable discovery</div>	

Manage support package

Purchase support package

Donate to Weblate

Activate support package ⓘ

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

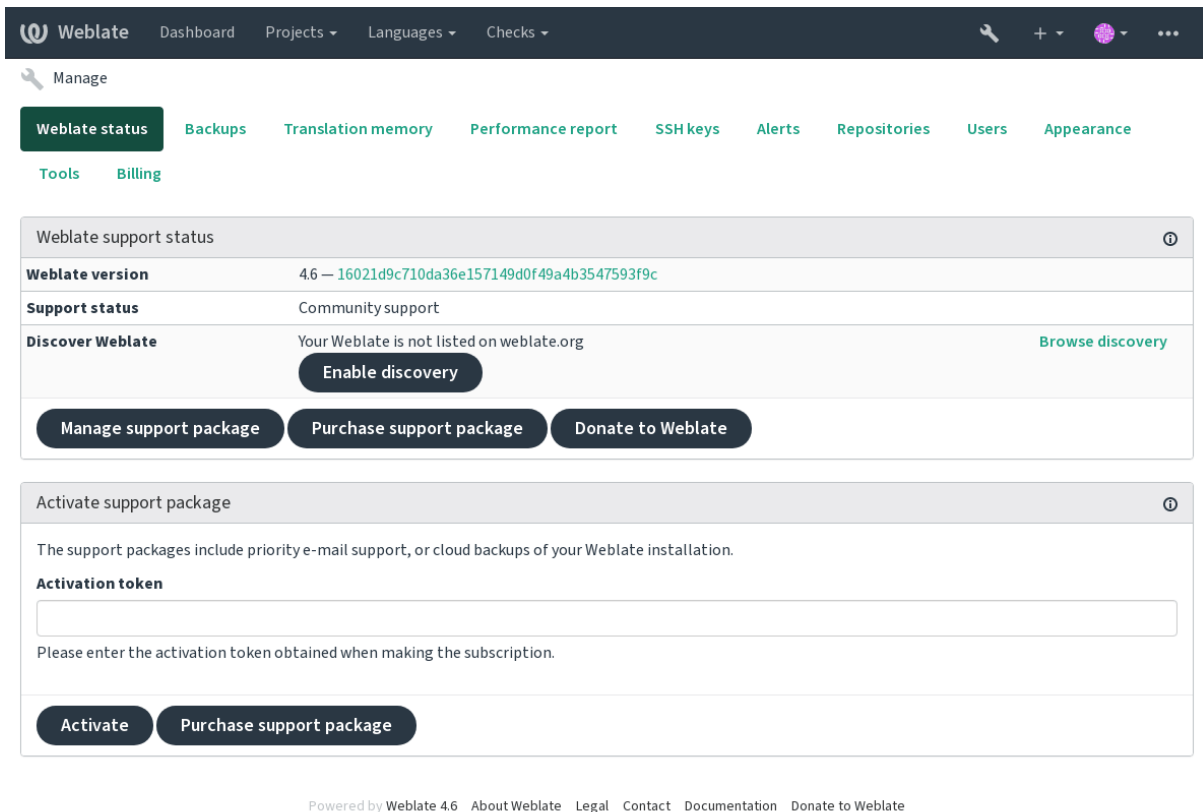
Activate

Purchase support package

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

Listing your server without a support subscription in Discover Weblate:

1. Register yourself at <<https://weblate.org/user/>>
2. Register your Weblate server in the discovery database at <<https://weblate.org/subscription/discovery/>>
3. Confirm the service activation in your Weblate and turn on the discovery listing in your Weblate management page using *Enable discovery* button:



The screenshot shows the Weblate dashboard interface. At the top is a navigation bar with links to Dashboard, Projects, Languages, and Checks. Below this is a 'Manage' section with various tabs: Weblate status (selected), Backups, Translation memory, Performance report, SSH keys, Alerts, Repositories, Users, and Appearance. The 'Weblate status' section displays the following information:

- Weblate version:** 4.6 — 16021d9c710da36e157149d0f49a4b3547593f9c
- Support status:** Community support
- Discover Weblate:** Your Weblate is not listed on weblate.org. There is a button 'Enable discovery' and a link 'Browse discovery'.
- Buttons: 'Manage support package', 'Purchase support package', and 'Donate to Weblate'.

Below the status section is the 'Activate support package' section, which includes a text input for an 'Activation token' and an 'Activate' button. The text below the input says: 'Please enter the activation token obtained when making the subscription.'

At the bottom of the dashboard, there is a footer with links: 'Powered by Weblate 4.6', 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Customizing listing

You can customize the listing by providing a text and image (570 x 260 pixels) at <https://weblate.org/user/>.

2.25 Documentos legais

Nota: Aqui você encontrará várias informações legais que você 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 sua responsabilidade garantir que 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 de 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 usuários finais são responsáveis por garantir tal conformidade.

Hosted Weblate serviço 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 Controles de criptografia dos EUA

O Weblate não contém nenhum código criptográfico, mas pode ser objeto de controles 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, ele irá incluir todas 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 em um grupo multiplicativo de um campo finito de tamanho maior do que 512 bits para um algoritmo assimétrico
- Logaritmos discretos em um 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 uma maneira onde nenhum código de criptografia estaria envolvido. Os recursos criptográficos incluem:

- Acessar servidores remotos usando protocolos seguros (HTTPS)
- Gerar assinaturas para commits de código (PGP)

Ver também:

Controles de Exportação (EAR) em Software de Código Aberto (*inglês*)

3.1 Contribuindo para o Weblate

Há dezenas de maneiras de melhorar o Weblate. Você pode escolher uma com a qual você se sente confortável, seja codificação, design gráfico, documentação, patrocínio ou uma ideia:

- *Relatando problemas no Weblate*
- *Começando a contribuir código para o Weblate*
- *Traduzindo o Weblate*
- *Contribua para a documentação do Weblate*
- *Discussões do Weblate*
- *Financiando o desenvolvimento do Weblate*

3.1.1 Traduzindo o Weblate

O Weblate está sendo continuamente [traduzido](#) usando o próprio Weblate. Sinta-se à vontade para participar do esforço de disponibilizar o Weblate no maior número possível de idiomas humanos. Isso traz o Weblate para mais perto de seus usuários!

Se você encontrar um possível erro no texto fonte, pode marcá-lo com um comentário no editor Weblate. Dessa forma, pode ser discutido e corrigido. Se você tiver certeza, também pode clicar no link na seção *Localização do texto fonte* e enviar um PR com sua correção.

3.1.2 Contribua para a documentação do Weblate

Você é bem-vindo para melhorar a página de documentação de sua escolha. Faça isso facilmente clicando 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. Você está escrevendo documentos técnicos, não um poema. Nem todos os leitores de documentos são falantes nativos, fique atento.
3. Não tenha medo de perguntar se você não tem certeza. Se você tiver que perguntar sobre algum recurso durante a edição, não altere seus documentos antes de ter a resposta. Isso significa: você muda ou pergunta. Não faça os dois ao mesmo tempo.
4. Verifique suas alterações executando 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 você 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: Você pode [traduzir os documentos](#).

3.1.3 Discussões do Weblate

Se você tiver uma ideia e não tiver certeza se ela é adequada para um problema, não se preocupe. Você pode entrar na comunidade de [discussões do GitHub](#).

3.1.4 Financiando o desenvolvimento do Weblate

Você pode impulsionar o desenvolvimento do Weblate na [página de doação](#). Os fundos coletados lá são usados para permitir hospedagem de grátiis 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 você 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

Gostaria de estar na lista? Veja as opções no [Doar para o Weblate](#).

3.2 Começando a contribuir código para o Weblate

Entenda o código-fonte do Weblate passando por *Código-fonte do Weblate*, *Frontend do Weblate* e *Componentes internos do Weblate*.

3.2.1 Começando com a base de código

Familiarize-se com a base de código do Weblate, dando uma olhada nos bugs etiquetados como [good first issue](#).

3.2.2 Executando Weblate localmente

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, você 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, você também pode querer 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
```

Ver também:

Instalando a partir do código-fonte

3.2.3 Executando Weblate localmente no Docker

Se você tiver Docker e docker-compose instalados, você pode colocar para funcionar o ambiente de desenvolvimento simplesmente executando:

```
./rundev.sh
```

Isso irá criar uma imagem do Docker de desenvolvimento e iniciá-lo. Weblate está sendo executado em <http://127.0.0.1:8080/> e você pode entrar como o usuário `admin`, usando `admin` como a senha. A nova instalação está vazia, então você pode querer seguir *Adicionando projetos e componentes de tradução*.

Os arquivos `Dockerfile` e `docker-compose.yml` para isso estão localizados no diretório `dev-docker`.

O script também aceita alguns parâmetros; para executar testes, execute-o com o parâmetro `test` e então especifique qualquer parâmetro `test`, por exemplo:

```
./rundev.sh test --failfast weblate.trans
```

Nota: Tenha cuidado para que os seus contêineres Docker estejam prontos e em execução antes de fazer os testes. Você pode verificar isso executando o comando `docker ps`.

Para exibir os logs:

```
./rundev.sh logs
```

Para parar os contêineres em segundo fundo, execute:

```
./rundev.sh stop
```

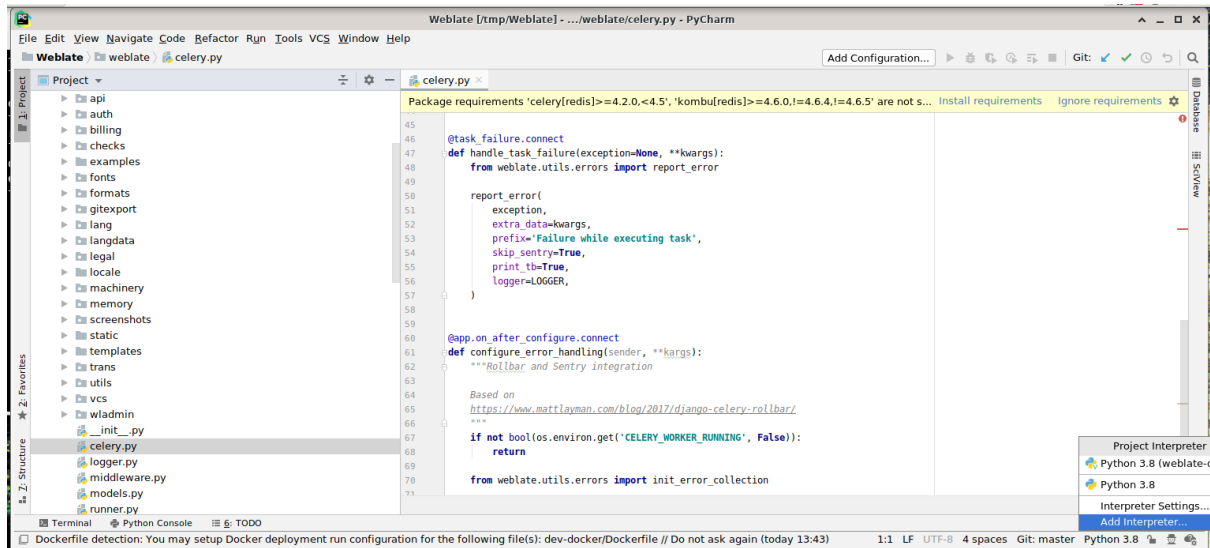
A execução do script sem argumentos vai recriar o contêiner Docker e reiniciá-lo.

Nota: Esta não é uma configuração adequada para a produção, pois inclui vários hacks que são inseguros, mas que tornam o desenvolvimento mais fácil.

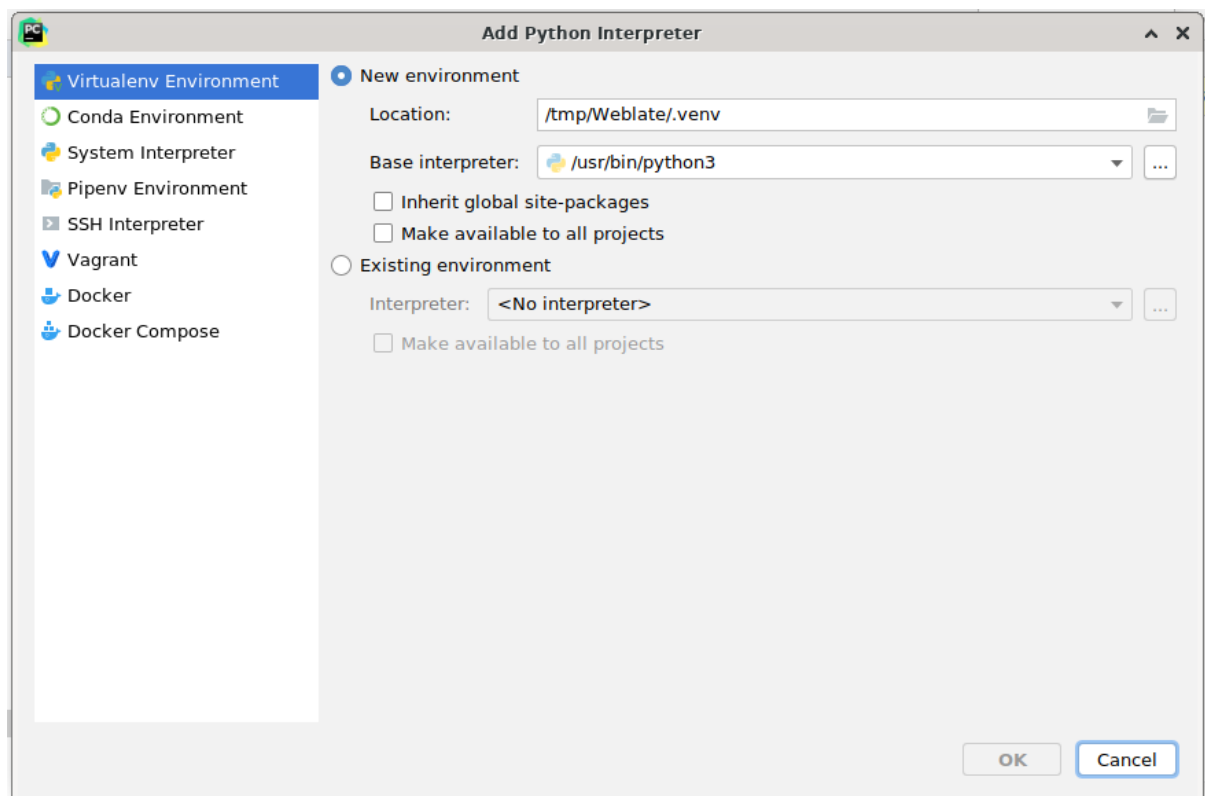
3.2.4 Codificando Weblate com PyCharm

PyCharm é uma IDE conhecida para Python, aqui estão algumas diretrizes para lhe ajudar a configurar seu projeto Weblate nele.

Considerando que você 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 você deseja:

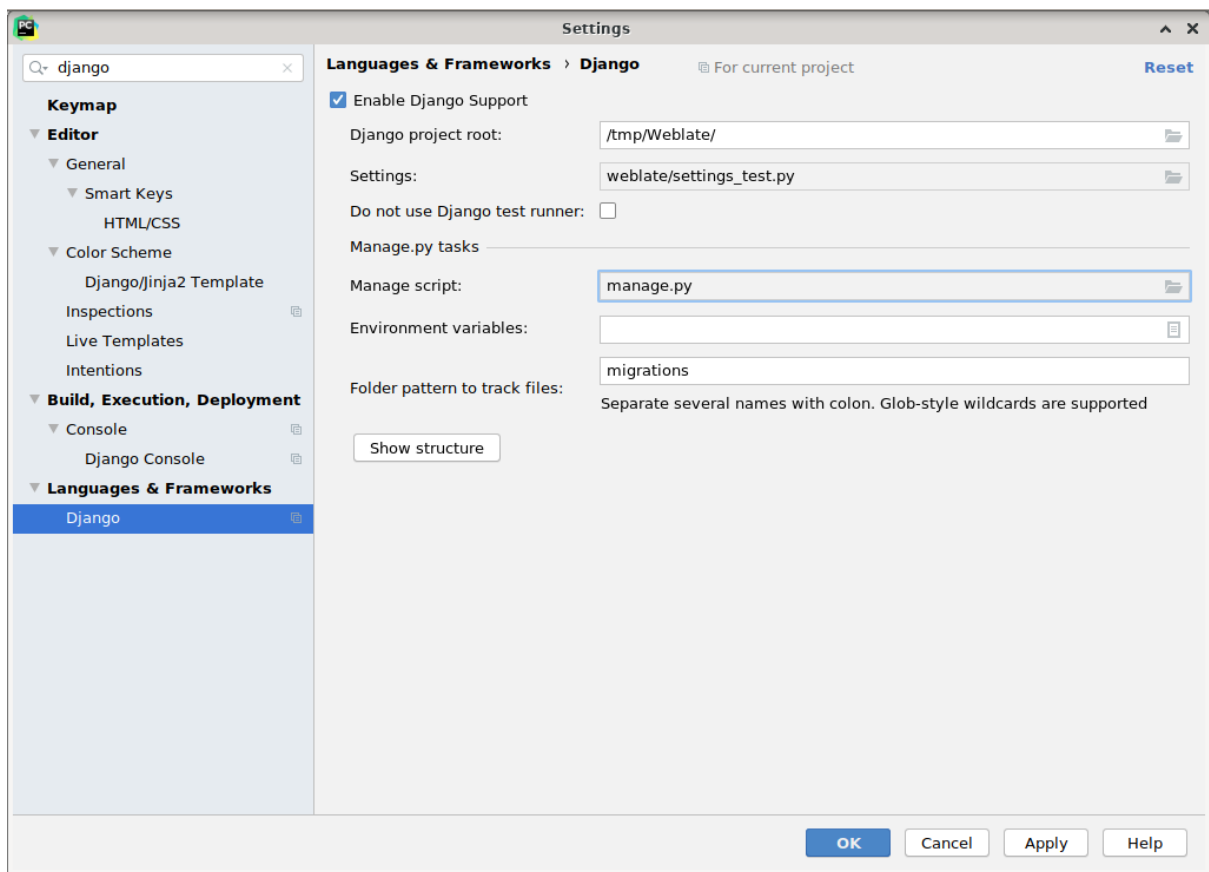


Você pode optar por deixar PyCharm criar o virtualenv para você, ou selecionar um já existente:



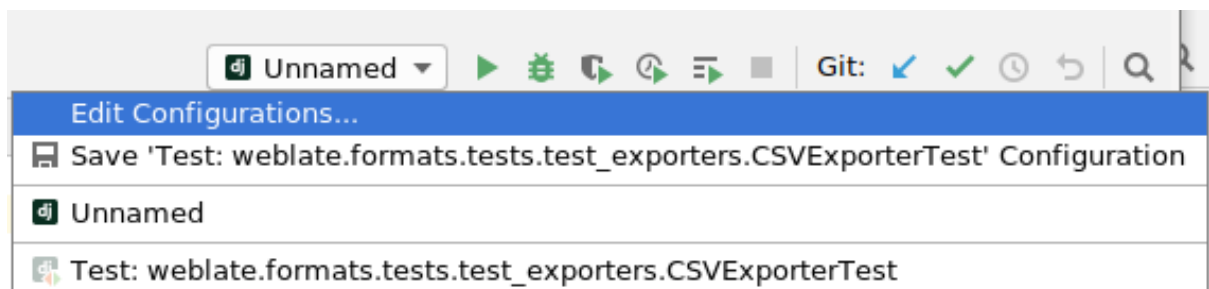
Não se esqueça de instalar as dependências quando o interpretador estiver configurado: faça isso através do console (o console do IDE vai usar diretamente o seu virtualenv por padrão, ou através da interface quando receber um aviso sobre dependências em falta).

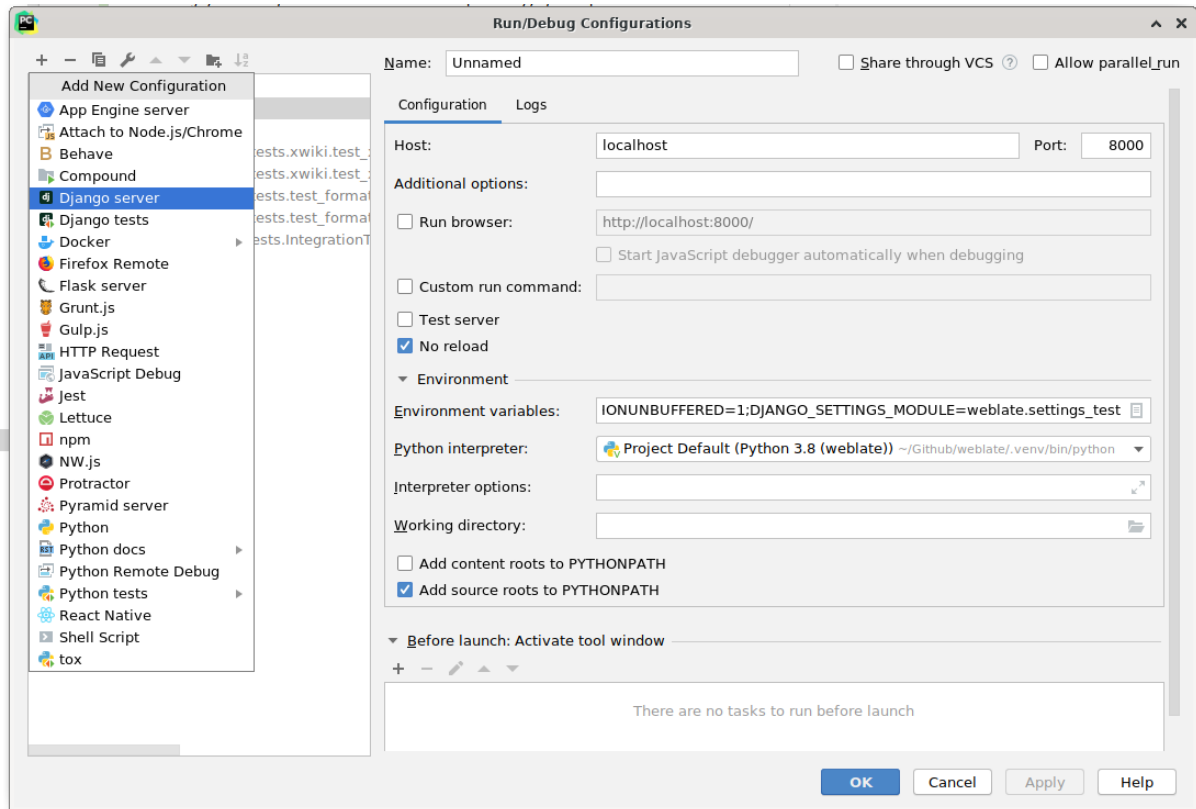
O segundo passo é definir a informação certa para usar o Django nativo dentro do PyCharm: a ideia é ser capaz de acionar imediatamente os testes unitários no IDE. Para isso, você precisa especificar o caminho raiz do projeto Django e o caminho para suas configurações:



Tenha cuidado, *Django project root* é a raiz real do repositório, não o subdiretório do Weblate. Sobre as configurações, você poderia usar o `weblate/settings_test.py` do repositório, mas você pode criar sua própria configuração e configurá-la lá.

O último passo é executar o servidor e colocar pontos de interrupção no código para poder depurá-lo. Isto é feito criando uma nova configuração do *Django Server*:





Dica: Tenha cuidado com a propriedade chamada *No reload*: ela evita do servidor ser recarregado automaticamente quando você modificar os arquivos. Isso permite que os pontos de interrupção de depuração existentes persistam, quando eles normalmente seriam descartados no recarregamento do servidor.

3.2.5 Inicializando sua instância de desenvolvimento

Você pode querer usar `import_demo` para criar traduções de demonstração e `createadmin` para criar um usuário administrador.

3.3 Código-fonte do Weblate

O Weblate é desenvolvido no [GitHub](#). Você é bem-vindo para criar um fork do código e abrir pull requests. Patches em qualquer outra forma são bem-vindos também.

Ver 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 usando o formatador de código **black**.

Para verificar a qualidade do código, você pode usar **flake8**, os plugins recomendados estão listados em `.pre-commit-config.yaml` e sua configuração é colocada 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 arquivos do commit estão sãos. Depois de instalá-lo (ele já está incluído no `requirements-lint.txt`), ative-o executando `pre-commit install` na sua cópia do Weblate. Desta forma, todas as suas alterações serão verificadas automaticamente.

Você também pode acionar a verificação manualmente, para verificar todos os arquivos executados:

```
pre-commit run --all
```

3.4 Depuração do Weblate

Os bugs podem se comportar como falhas no aplicativo ou como um mau comportamento. Você é bem-vindo a coletar informações sobre qualquer problema desse tipo e enviá-las para o [rastreador 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.

Ver também:

Desabilitar o modo de depuração

3.4.2 Logs do Weblate

Weblate pode produzir registros (logs) detalhados do que está acontecendo em segundo fundo. Na configuração padrão, ele usa syslog e isso faz o log aparecer em `/var/log/messages` ou `/var/log/syslog` (dependendo da configuração do seu daemon syslog).

O processo de Celery (veja *Tarefas de fundo usando Celery*) normalmente produz também registros próprios. O exemplo de configurações de todo o sistema registra em vários arquivos em `/var/log/celery/`.

Os contêineres Docker registram a sua saída (como é habitual no mundo Docker), por isso pode olhar para os logs utilizando `docker-compose logs`.

Ver também:

Amostra de configuração contém uma configuração de `LOGGING`.

3.4.3 Não processando 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 usando Celery*
- Verifique o status da fila do Celery em *Interface de gerenciamento* ou use `celery_queues`
- Procure erros nos registros do Celery (veja *Logs do Weblate*)

3.4.4 Não recebendo e-mails do Weblate

Você pode verificar se o e-mail de saída está funcionando corretamente usando o comando de gerenciamento `send-testemail` (veja *Invocando comandos de gerenciamento* para instruções sobre como invocá-lo em diferentes ambientes) ou usando *Interface de gerenciamento* sob a aba *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 processando tarefas de segundo plano* para depuração disso.

3.4.5 Analisando falhas de aplicativo

Caso o aplicativo falhe, é útil recolher o máximo de informação possível sobre a falha. A maneira mais fácil de conseguir isso é usando serviços de terceiros que podem coletar tais informações automaticamente. Você 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 usuário, 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 registros relevantes mostrando 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, você 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 de sua estrutura de código 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 usando [Sphinx](#).

dev-docker Código do Docker para executar o servidor de desenvolvimento, veja [Executando Weblate localmente no Docker](#).

weblate Código-fonte do Weblate como um aplicativo Django, veja [Componentes internos do Weblate](#).

weblate/static Arquivos de cliente (CSS, Javascript e imagens), veja [Frontend do Weblate](#).

3.5.2 Módulos

O Weblate consiste em vários aplicativos Django (alguns opcionais, veja [Módulos opcionais do Weblate](#)):

`accounts`

Conta de usuário, perfis e notificações.

`addons`

Extensões para ajustar o comportamento do Weblate, veja [Extensões](#).

`api`

API baseada no [framework Django REST](#).

`auth`

Autenticação e permissões.

`billing`

O módulo opcional de [Cobrança](#).

`checks`

Módulo de [Verificações de qualidade](#) de texto de tradução.

`fonts`

Módulo de verificações de renderização de fonte.

`formats`

Camada de abstração de formato de arquivo 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`

Gerenciamento de capturas de tela e módulo OCR.

`trans`

Módulo principal que trata das traduções.

`utils`

Vários utilitários auxiliares.

`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*)

Verifica se a extensão é compatível com o componente dado.

configure (*settings*)

Salva a configuração.

daily (*component*)

Gancho acionado diariamente.

classmethod `get_add_form` (*user, component, **kwargs*)

Retorna um formulário de configuração para adicionar uma nova extensão.

get_settings_form (*user, **kwargs*)

Retorna um formulário de configuração para esta extensão.

post_add (*translation*)

Gancho acionado após nova tradução ser adicionada.

post_commit (*component*)

Gancho acionado após alterações terem seu commit feito 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*) – Se a operação de extensão deve ignorar o push de alterações para o upstream. Normalmente você pode passar isso para métodos subjacentes como `commit_and_push` ou `commit_pending`.

pre_commit (*translation, author*)

Gancho acionado antes das alterações terem seu commit feito para o repositório.

pre_push (*component*)

Gancho acionado antes do repositório ter seu push feito para o upstream.

pre_update (*component*)

Hook acionado antes do repositório ser atualizado a partir do upstream.

save_state ()

Salva informações do estado da extensão.

stay_on_create = False

Classe base para as extensões do Weblate.

store_post_load (*translation, store*)

Gancho acionado após um arquivo ser analisado.

Ele recebe uma instância de uma classe de formato de arquivo como um argumento.

Isso é útil para modificar os parâmetros de classe de formato de arquivo, por exemplo, ajustar como o arquivo 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 addon should receive
    events = (EVENT_PRE_COMMIT,)
    # Addon unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example addon")
    # Detailed addon description
    description = _("This addon does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Frontend do Weblate

O frontend atualmente é construído usando 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 Gerenciamento de dependências

O gerenciador 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 arquivos JavaScript e CSS.

Também usamos [ESLint](#) para verificar o código JavaScript.

3.7.4 Localização

Se você precisar de qualquer texto visível para o usuário no código do frontend, ele deve ser localizável. Na maioria dos casos, tudo que você precisa é envolver 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'
```

Ver também:

[Tópico sobre tradução na documentação do Django](#)

3.7.5 Ícones

Weblate atualmente usa ícones de design de material. Caso você esteja procurando 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 você tenha ou sugerir melhorias para o Weblate lá. Existem vários modelos preparados para lhe orientar confortavelmente durante o relatório de problemas.

Se o que você encontrou é um problema de segurança no Weblate, consulte a seção [Problemas de segurança](#) abaixo.

Se você não tem certeza sobre seu relatório de bug ou solicitação de recurso, você 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 você 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 você não quiser usar o HackerOne, por qualquer motivo, você pode enviar o relatório por e-mail para mi-chal@cihar.com. Você pode optar por criptografá-lo usando esta chave PGP `3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D`. Você também pode obter a chave PGP em [Keybase](#).

Nota: O Weblate depende de componentes de terceiros para muitas coisas. Caso você encontre uma vulnerabilidade afetando 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 adicionando casos de teste para qualquer nova funcionalidade e verifique se funciona.

3.9.1 Integração contínua

Os resultados do teste atual podem ser encontrados em [GitHub Actions](#) e a cobertura é relatada em [Codecov](#).

Existem vários trabalhos para verificar diferentes aspectos:

- Testes de unidade
- Construção de documentação e links externos
- Testes de migração de todas as versões com suporte
- Linting de código
- Verificação de configuração (garante que os arquivos 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 definindo o arquivo 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: Você precisará 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 de 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 usando variáveis de ambiente:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}
```

(continua na próxima página)

(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, você deve coletar arquivos estáticos, pois alguns testes contam com a presença deles:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

Você 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 contêiner Docker do desenvolvedor, consulte [Executando Weblate localmente no Docker](#).

Ver 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 arquivos JSON externos.

3.10.1 Esquema de memória de tradução do Weblate

https://weblate.org/schemas/weblate-memory.schema.json	
tipo	<i>array</i>
items	<i>O item da memória de tradução</i>
	tipo
	<i>objeto</i>
	propriedades
	• categoria
	<i>A Categoria do Fio da Meada</i>
	1 is global, 2 is shared, 10000000+ are project specific, 20000000+ are user specific
	tipo
	<i>integer</i>
	exemplos
	1
	minimum
	0
	padrão
	1
	• origem
	<i>The String Origin</i>
	Nome do arquivo ou nome de componente
	tipo
	<i>Fio da Meada</i>
	exemplos
	test.tmx
	project/component
	padrão
	• source
	<i>O Fio da Meada fonte</i>
	tipo
	<i>Fio da Meada</i>
	exemplos
	Olá
	minLength
	1
	padrão
	• Idioma_fonte
	<i>O Idioma Fonte</i>

continua na próxima página

Tabela 1 – continuação da página anterior

		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		tipo	<i>Fio da Meada</i>
		exemplos	Inglês
		pattern	^[^]+\$
		padrão	
	• target	<i>O Fio da Meada alvo</i>	
		tipo	<i>Fio da Meada</i>
		exemplos	Ahoj
		minLength	1
		padrão	
	• idioma_alvo	<i>O Idioma Alvo</i>	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		tipo	<i>Fio da Meada</i>
		exemplos	cs
		pattern	^[^]+\$
		padrão	
	additionalProperties		False
definições			

Ver também:

Memória de tradução, dump_memory, import_memory

3.10.2 Exportação de dados de usuário do Weblate

https://weblate.org/schemas/weblate-userdata.schema.json			
tipo	<i>objeto</i>		
propriedades			
• basic	<i>Básico</i>		
	tipo	<i>objeto</i>	
	propriedades		
	• Nome de usuário	<i>Nome de usuário</i>	
		tipo	<i>Fio da Meada</i>
		exemplos	admin
		padrão	
	• Nome_completo	<i>Nome completo</i>	
		tipo	<i>Fio da Meada</i>
		exemplos	Administração Weblate
		padrão	
	• email	<i>E-mail</i>	
		tipo	<i>Fio da Meada</i>
		exemplos	noreply@example.com
		padrão	
	• Data_juntado	<i>Data juntado</i>	
		tipo	<i>Fio da Meada</i>
		exemplos	2019-11-18T18:53:54.862Z
		padrão	
• Perfil	<i>Perfil</i>		
	tipo	<i>objeto</i>	
	propriedades		
	• Idioma	<i>Idioma</i>	
		tipo	<i>Fio da Meada</i>
		exemplos	cs
		pattern	^.*\$
		padrão	

continua na próxima página

Tabela 2 – continuação da página anterior

	• sugerido	Número de Fios da Meada sugeridos	
		tipo	integer
		exemplos	1
		padrão	0
	• traduzido	Número de Fios da Meada traduzidos	
		tipo	integer
		exemplos	24
		padrão	0
	• Upado	Número de prints de tela upados	
		tipo	integer
		exemplos	1
		padrão	0
	• hide_completed	Ocultar as traduções concluídas no painel	
		tipo	boolean
		exemplos	False
		padrão	True
	• secondary_in_zen	Mostrar as traduções secundárias no modo Zen	
		tipo	boolean
		exemplos	True
		padrão	True
	• hide_source_secondary	Ocultar a fonte caso uma tradução secundária exista	
		tipo	boolean
		exemplos	False
		padrão	True
	• link_editor	Link do editor	
		tipo	Fio da Meada
		exemplos	
		pattern	^.*\$
		padrão	
	• Modo_traduzir	Modo do editor de tradução	
		tipo	integer
		exemplos	0
		padrão	0
	• zen_mode	Modo editor Zen	
		tipo	integer
		exemplos	0
		padrão	0
	• Caracteres especiais	Caracteres especiais	
		tipo	Fio da Meada
		exemplos	
		pattern	^.*\$
		padrão	
	• Visão_painel	Visão do painel	
		tipo	integer
		exemplos	1
		padrão	0
	• dashboard_components_list	Lista de componentes padrão	
		tipo	null
		anyOf	tipo
		tipo	integer
	• Idiomas	Idiomas traduzidos	
		tipo	array
		padrão	
		items	Código do idioma
		tipo	Fio da Meada
		exemplos	cs

continua na próxima página

Tabela 2 – continuação da página anterior

			pattern	^.*\$		
			padrão			
	• Idiomas secundários	<i>Idiomas secundários</i>				
		tipo	array			
		padrão				
		items	<i>Código do idioma</i>			
			tipo	Fio da Meada		
			exemplos	sk		
			pattern	^.*\$		
			padrão			
	• Observados	<i>Projetos observados</i>				
		tipo	array			
		padrão				
		items	<i>Projeto Lesma</i>			
			tipo	Fio da Meada		
			exemplos	Weblate		
			pattern	^.*\$		
			padrão			
• Registro de auditoria	<i>Registro de auditoria</i>					
	tipo	array				
	padrão					
	items	<i>Items</i>				
		tipo	objeto			
		propriedades				
		• Endereço	<i>Endereço de IP</i>			
			tipo	Fio da Meada		
			exemplos	127.0.0.1		
			pattern	^.*\$		
			padrão			
		• Agente_usuario	<i>Agente de usuário</i>			
			tipo	Fio da Meada		
			exemplos	PC / Linux / Firefox 70.0		
			pattern	^.*\$		
			padrão			
		• timestamp	<i>Timestamp</i>			
			tipo	Fio da Meada		
exemplos			2019-11-18T18:58:30.845Z			
pattern			^.*\$			
padrão						
• Atividade		<i>Atividade</i>				
		tipo	Fio da Meada			
	exemplos	entrar				
	pattern	^.*\$				
	padrão					
definições						

Ver também:*Perfil do usuário, 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 - você sempre deve atualizar para x.0 antes de atualizar para versões x.y superiores.

Ver também:

Atualizando o Weblate

3.11.2 Planejamento de lançamento

Os recursos para os próximos lançamentos são coletados usando marcos do GitHub, você pode ver nosso roteiro em [<https://github.com/WeblateOrg/weblate/milestones>](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 usando `./scripts/list-translated-languages`.
2. Definir a versão final usando `./scripts/prepare-release`.
3. Certificar-se de que as capturas de tela estejam atualizadas usando `make -C docs update-screenshots`.
4. Junte todas as traduções pendentes usando `wlc push; git remote update; git merge origin/weblate`

Fazer o lançamento:

5. Criar um lançamento usando `./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 nos links de download do site.
12. Incrementar a versão no repositório usando `./scripts/set-version`.

Para criar tags usando o script `./scripts/create-release`, você 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 download Weblate (os downloads do site são copiados lá)

3.12 Segurança e privacidade

Dica: Na Weblate, a segurança mantém um ambiente que valoriza a privacidade de nossos usuários.

O desenvolvimento do Weblate segue as [Melhores Práticas da Core Infrastructure Initiative](#) da Linux Foundation.

Ver também:

Problemas de segurança

3.12.1 Rastreamento de dependências para vulnerabilidades

Problemas de segurança em nossas dependências são monitorados usando [Dependabot](#). Isso cobre as bibliotecas Python e JavaScript, e a versão estável mais recente tem suas dependências atualizadas 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 contêiner Docker

Os contêineres Docker são verificados usando [Anchore](#) e [Trivy](#).

Isso nos permite detectar vulnerabilidades antecipadamente e lançar melhorias rapidamente.

Você pode obter os resultados dessas varreduras no GitHub – eles são armazenados como artefatos em nosso CI no formato SARIF (Static Analysis Results Interchange Format).

Ver também:

Integração contínua

3.13 Sobre o Weblate

3.13.1 Objetivos do projeto

Ferramenta de localização contínua baseada na web com *Integração com controle de versão* suportando uma ampla gama de *Formatos de arquivos 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 do projeto

A página inicial é <https://weblate.org> e há um serviço hospedado na nuvem em <https://hosted.weblate.org>. Esta documentação pode ser encontrada em <https://docs.weblate.org>.

3.13.4 Logotipos do projeto

Os logotipos do projeto e outros gráficos estão disponíveis no repositório <<https://github.com/WeblateOrg/graphics/>>.

3.13.5 Liderança

Este projeto é mantido por Michal Čihař <michal@cihar.com>.

3.13.6 Autores

Weblate foi iniciado por Michal Čihař <michal@cihar.com>. Desde sua criação, em 2012, milhares de pessoas contribuíram.

3.14 Licença

Copyright (C) 2012 - 2021 Michal Čihař <michal@cihar.com>

Este programa é um software livre: você 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 (a 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.

Você 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 4.6.1

Released on May 2nd 2021.

- Remove obsolete spam protection code.
- Improve source plural check accuracy.
- Update list of user interface languages in Docker.
- Improved error messages when creating pull requests.
- Fixed creating pull requests on Pagure.
- Fixed triggering automatically installed add-ons.
- Fixed possible caching issues on upgrade.
- Fixed adding new units to monolingual translations using upload.

[All changes in detail.](#)

4.2 Weblate 4.6

Released on April 19th 2021.

- The `auto_translate` management command has now a parameter for specifying translation mode.
- Added support for *Arquivos texto*.
- Added trends and metrics for all objects.
- Added support for direct copying text from secondary languages.
- Added date filtering when browsing changes.
- Improved activity charts.
- Sender for contact form e-mails can now be configured.
- Improved parameters validation in component creation API.
- The rate limiting no longer applies to superusers.

- Improved automatic translation addon performance and reliability.
- The rate limiting now can be customized in the Docker container.
- API for creating components now automatically uses *URLs internas do Weblate*.
- Simplified state indication while listing strings.
- Password hashing now uses Argon2 by default.
- Barras de progresso simplificadas indicando o status da tradução.
- Renamed *Adiciona idiomas em falta* to clarify the purpose.
- Corrigido o salvamento do estado de texto para XLIFF.
- Adicionada pesquisa para todo idioma.
- Initial support for *Scaling horizontally* the Docker deployment.

Todas as alterações em detalhes.

4.3 Weblate 4.5.3

Released on April 1st 2021.

- Fixed metrics collection.
- Fixed possible crash when adding strings.
- Melhorados os exemplos de consulta de pesquisa.
- Fixed possible loss of newly added strings on replace upload.

4.4 Weblate 4.5.2

Released on March 26th 2021.

- Agendamento configurável para tradução automática.
- Added Lua format check.
- Ignore format strings in the *Há palavras duplicadas de forma consecutiva* 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.
- Improved performance when adding new strings.
- Several bug fixes in quality checks.
- Several performance improvements.
- Adicionada integração com *Descubra Weblate*.
- Fixed checks behavior with read-only strings.

Todas as alterações em detalhes.

4.5 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.
- Better handling for uploads of monolingual PO using API.
- Improved alerts behavior glossaries.
- Improved Markdown link checks.
- Indicate glossary and source language in breadcrumbs.
- Listagem paginada de componentes de grandes projetos.
- Improved performance of translation, component or project removal.
- Improved bulk edit performance.
- Fixed preserving “Needs editing” and “Approved” states for ODF files.
- Interface melhorada para personalização de downloads de arquivos de tradução

Todas as alterações em [detalhes](#).

4.6 Weblate 4.5

Released on February 19th 2021.

- Suporte adicionado para `lua-format` usado em gettext PO.
- Added support for sharing a component between projects.
- Fixed multiple unnamed variables check behavior with multiple format flags.
- Dropped mailing list field on the project in favor of generic instructions for translators.
- Added pseudolocale generation addon.
- Added support for TermBase eXchange files.
- Adicionado suporte para definir manualmente variantes de texto usando um marcador.
- Improved performance of consistency checks.
- Improved performance of translation memory for long strings.
- Added support for searching in explanations.
- Strings can now be added and removed in bilingual formats as well.
- Extend list of supported languages in Amazon Translate machine translation.
- Habilita automaticamente verificações de MessageFormat para Properties do Java.
- Added a new upload method to add new strings to a translation.
- Added a simple interface to browse translation.
- 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.
- Added support for non-translatable or forbidden terms in the glossary.

- Added support for defining terminology in a glossary.
- Moved text direction toggle to get more space for the visual keyboard.
- Adicionada a opção de observar automaticamente os projetos para os quais o usuário contribuiu.
- Added check whether translation matches the glossary.
- Added support for customizing navigation text color.

[Todas as alterações em detalhes.](#)

4.7 Weblate 4.4.2

Released on January 14th 2021.

- Fixed corruption of one distributed MO file.

4.8 Weblate 4.4.1

Released on January 13th 2021.

- Fixed reverting plural changes.
- Fixed displaying help for project settings.
- Improved administration of users.
- Improved handling of context in monolingual PO files.
- Fixed cleanup addon 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 Pagine.
- Improved performance when displaying ghost translations.
- Reimplemented translation editor to use native browser textarea.
- Fixed cleanup addon breaking adding new strings.
- Added API for addons.

[Todas as alterações em detalhes.](#)

4.9 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.

- Adicionada extensão para remover textos vazios de arquivos de tradução.
- 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.
- O componente contém arquivo de tradução para o idioma fonte.
- 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.](#)

4.10 Weblate 4.3.2

Released on November 4th 2020.

- Fixed crash on certain component filemasks.
- Improved accuracy of the consecutive duplicated words check.
- Added support for Pagure pull requests.
- 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.](#)

4.11 Weblate 4.3.1

Released on October 21st 2020.

- Melhorada o desempenho da tradução automática.
- Expiração de sessão fixa para usuários autenticados.
- Adicionar suporte para ocultar informação de versão.
- 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.](#)

4.12 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.
- Corrigidas a contagem das estatísticas após remoção de sugestão.
- Estendido perfil público de usuário.
- Fixed configuration of enforced checks.
- Improve documentation about built-in backups.
- Moved source language attribute from project to a component.
- Adicionada verificação de formatação Vue I18n.
- Generic placeholders check now supports regular expressions.
- Improved look of Matrix mode.
- Maquinaria agora é chamada de 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.
- Melhorado o rastreamento de textos fontes.
- Improved JSON, YAML and CSV formats compatibility.
- Suporte adicionado para remover Fios da Meada.
- Improved performance of file downloads.

- Improved repository management view.
- Automatically enable java-format for Android.
- Suporte adicionado para prints da tela localizados.
- Suporte adicionado para Python 3.9.
- Fixed translating HTML files under certain conditions.

Todas as alterações em detalhes.

4.13 Weblate 4.2.2

Released on September 2nd 2020.

- Corrigido correspondência de textos fonte para formatos JSON.
- Fixed login redirect for some authentication configurations.
- Corrigida autenticação por LDAP com sincronização de grupo.
- Corrigida falha em relatar progresso das traduções automáticas.
- Fixed Git commit squashing with trailers enabled.
- Fixed creating local VCS components using API.

4.14 Weblate 4.2.1

Released on August 21st 2020.

- Fixed saving plurals for some locales in Android resources.
- Fixed crash in the cleanup addon for some XLIFF files.
- Allow setting up localization CDN in Docker image.

4.15 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.
- Adicionar suporte para bloquear componentes que sofrem erros de repositório.
- Simplified main navigation (replaced buttons with icons).
- Improved language code handling in Google Translate integration.
- The Git squash addon 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.
- Adicionada verificação para literais de modelo de ECMAScript.
- Adicionada opção para observar um componente.
- Removed leading dot from JSON unit keys.
- Removida fila separada de Celery para memória de tradução.
- Permite traduzir todos os componentes um idioma de uma só vez.
- Allow to configure Content-Security-Policy HTTP headers.
- Added support for aliasing languages at project level.
- New addon to help with HTML or JavaScript localization, see *CDN de localização JavaScript*.
- The Weblate domain is now configured in the settings, see *SITE_DOMAIN*.
- Adiciona suporte para pesquisar componente e projeto.

4.16 Weblate 4.1.1

Released on June 19th 2020.

- Fixed changing autofix or addons 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*.

4.17 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: *Textos do PHP de Laravel*, *Arquivos HTML*, *Formato OpenDocument*, *Formato IDML*, *Arquivos RC do Windows*, *Traduções de INI*, *Traduções de Inno Setup INI*, *Propriedades GWT*, *Arquivos JSON go-i18n*, *Arquivo ARB*.
- Consistently use dismissed as state of dismissed checks.
- Add support for configuring default addons 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 a muito tempo*, *Há palavras duplicadas de forma consecutiva*.
- 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 usando marcadores*.
- Always show rendered text check if enabled.
- API now supports filtering of changes.
- Added support for sharing glossaries between projects.

4.18 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.

4.19 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.

4.20 Weblate 4.0.2

Released on April 27th 2020.

- Improved performance of translation stats.
- Improved performance of changing labels.
- Improved bulk edit performance.
- Improved translation memory performance.
- Fixed possible crash on component deletion.
- Fixed displaying of translation changes in some corner cases.
- Improved warning about too long celery queue.
- Fixed possible false positives in the consistency check.
- Fixed deadlock when changing linked component repository.
- Included edit distance in changes listing and CSV and reports.
- Avoid false positives of punctuation spacing check for Canadian French.
- Fixed XLIFF export with placeholders.
- Fixed false positive with zero width check.
- Improved reporting of configuration errors.
- Fixed bilingual source upload.

- Automatically detect supported languages for DeepL machine translation.
- Fixed progress bar display in some corner cases.
- Fixed some checks triggering on non translated strings.

4.21 Weblate 4.0.1

Released on April 16th 2020.

- Fixed package installation from PyPI.

4.22 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.
- Exibe verificações com falha em 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 não alterada* avoid internal blacklist.
- Improved comments extraction from monolingual po files.
- Renamed whiteboard messages to announcements.
- Fixed occasional problems with registration mails.
- Improved LINGUAS update addon to handle more syntax variants.
- Fixed editing monolingual XLIFF source file.
- Added support for exact matching in *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.

4.23 Weblate 3.x series

4.23.1 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 addons.
- 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.

4.23.2 Weblate 3.11.2

Released on February 22nd 2020.

- Fixed rendering of suggestions.
- Fixed some strings wrongly reported as having no words.

4.23.3 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.

4.23.4 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.
- Adicionado suporte para agrupar formas de textos.
- 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 addon.
- Added option for *Forçando 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 addons now do full sync of translations in Weblate.

4.23.5 Weblate 3.10.3

Released on January 18th 2020.

- Support for translate-toolkit 2.5.0.

4.23.6 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 addon to configure YAML formatting.
- Fixed same plurals check to not fire on single plural form languages.
- Fixed regex search on some fields.

4.23.7 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.

4.23.8 Weblate 3.10

Released on December 20th 2019.

- Improved application user interface.
- 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.

4.23.9 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.

4.23.10 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 addon to bootstrap translations.
- Extend automatic translation to add suggestions.
- Display addon parameters on overview.
- Sentry is now supported through modern Sentry SDK instead of Raven.
- Changed example settings to be better fit for production environment.
- Added automated backups using BorgBackup.
- Split cleanup addon for RESX to avoid unwanted file updates.
- Added advanced search capabilities.
- Allow users to download their own reports.
- Added localization guide to help configuring components.
- Added support for GitLab merge requests.
- Improved display of repository status.
- Perform automated translation in the background.

4.23.11 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 addons.
- 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 addons 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.

4.23.12 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.

4.23.13 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 addon.
- Delay opening SMTP connection when sending notifications.
- Improved error logging.
- Allow custom location in MO generating addon.
- Added addons to cleanup old suggestions or comments.
- Added option to enable horizontal mode in the Zen editor.
- Improved import performance with many linked components.
- Fixed examples installation in some cases.
- Improved rendering of alerts in changes.
- Added new horizontal stats widget.
- Improved format strings check on plurals.
- Added font management tool.
- New check for rendered text dimensions.
- Added support for subtitle formats.
- Include overall completion stats for languages.
- Added reporting at project and global scope.
- Improved user interface when showing translation status.
- New Weblate logo and color scheme.
- New look of bitmap badges.

4.23.14 Weblate 3.6.1

Released on April 26th 2019.

- Improved handling of monolingual XLIFF files.
- Fixed digest notifications in some corner cases.
- Fixed addon script error alert.
- Fixed generating MO file for monolingual PO files.
- Fixed display of uninstalled checks.
- Indicate administered projects on project listing.
- Allow update to recover from missing VCS repository.

4.23.15 Weblate 3.6

Released on April 20th 2019.

- Add support for downloading user data.
- Addons are now automatically triggered upon installation.
- Improved instructions for resolving merge conflicts.
- Cleanup addon is now compatible with app store metadata translations.
- Configurable language code syntax when adding new translations.

- Warn about using Python 2 with planned termination of support in April 2020.
- Extract special characters from the source string for visual keyboard.
- Extended contributor stats to reflect both source and target counts.
- Admins and consistency addons can now add translations even if disabled for users.
- Fixed description of toggle disabling Language–Team header manipulation.
- Notify users mentioned in comments.
- Removed file format autodetection from component setup.
- Fixed generating MO file for monolingual PO files.
- Added digest notifications.
- Added support for muting component notifications.
- Added notifications for new alerts, whiteboard messages or components.
- Notifications for administered projects can now be configured.
- Improved handling of three letter language codes.

4.23.16 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 addon 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.

4.23.17 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.
- Addon 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 addon now removes projects for unpaid billings after 45 days.

4.23.18 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 addon 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.

4.23.19 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.

4.23.20 Weblate 3.2.2

Released on October 20th 2018.

- Remove no longer needed Babel dependency.
- Updated language definitions.
- Improve documentation for addons, LDAP and Celery.
- Fixed enabling new dos-eol and auto-java-messageformat flags.
- Fixed running setup.py test from PyPI package.
- Improved plurals handling.
- Fixed translation upload API failure in some corner cases.
- Fixed updating Git configuration in case it was changed manually.

4.23.21 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.

4.23.22 Weblate 3.2

Released on October 6th 2018.

- Add install_addon management command for automated addon installation.
- Allow more fine grained ratelimit settings.
- Added support for export and import of Excel files.
- Improve component cleanup in case of multiple component discovery addons.
- 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.

4.23.23 Weblate 3.1.1

Released on July 27th 2018.

- Fix testsuite failure on some setups.

4.23.24 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.
- New addon to mark unchanged translations as needing edit.
- Add support for jumping to specific location while translating.
- Downloaded translations can now be customized.
- Improved calculation of string similarity in translation memory matches.
- Added support by signing Git commits by GnuPG.

4.23.25 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.

4.23.26 Weblate 3.0

Released on June 1st 2018.

- Rewritten access control.
- Several code cleanups that lead to moved and renamed modules.
- New addon for automatic component discovery.
- The `import_project` management command has now slightly different parameters.
- Added basic support for Windows RC files.
- New addon to store contributor names in PO file headers.
- The per component hook scripts are removed, use addons instead.
- Add support for collecting contributor agreements.
- Access control changes are now tracked in history.
- New addon to ensure all components in a project have same translations.
- Support for more variables in commit message templates.
- Add support for providing additional textual context.

4.24 Weblate 2.x series

4.24.1 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 addon to configure gettext output wrapping.
- New addon to configure JSON formatting.
- Added support for authentication in API using RFC 6750 compatible Bearer authentication.
- Added support for automatic translation using machine translation services.
- 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.

4.24.2 Weblate 2.19.1

Released on February 20th 2018.

- Fixed migration issue on upgrade from 2.18.
- Improved file upload API validation.

4.24.3 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 addons to customize translation workflows.

4.24.4 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.

4.24.5 Weblate 2.17.1

Released on October 13th 2017.

- Fixed running testsuite in some specific situations.
- Locales updates.

4.24.6 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.

4.24.7 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.

4.24.8 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.

4.24.9 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.

4.24.10 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.

4.24.11 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.

4.24.12 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.

4.24.13 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.

4.24.14 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.
- Melhor compatibilidade com várias versões git na extensão do exportador git.
- Included per project and component stats.
- Added language code mapping for better support of Microsoft Translate API.
- Moved fulltext cleanup to background job to make translation removal faster.
- Fixed displaying of plural source for languages with single plural form.
- Improved error handling in import_project.
- Various performance improvements.

4.24.15 Weblate 2.10.1

Released on Jan 20th 2017.

- Do not leak account existence on password reset form (CVE-2017-5537).

4.24.16 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.

4.24.17 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.

4.24.18 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.

4.24.19 Weblate 2.7

Released on Jul 10th 2016.

- Removed Google web translate machine translation.
- Improved commit message when adding translation.
- Fixed Google Translate API for Hebrew language.
- Compatibility with Mercurial 3.8.
- Added import_json management command.
- Correct ordering of listed translations.
- Show full suggestion text, not only a diff.
- Extend API (detailed repository status, statistics, ...).
- Testsuite no longer requires network access to test repositories.

4.24.20 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.

4.24.21 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.
- Corrigida a pesquisa 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.

4.24.22 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.

4.24.23 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.

4.24.24 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.

4.24.25 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.

4.24.26 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.

4.25 Weblate 1.x series

4.25.1 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.

4.25.2 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.

4.25.3 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.

4.25.4 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.

4.25.5 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.

4.25.6 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.

4.25.7 Weblate 1.3

Released on November 16th 2012.

- Compatibility with PostgreSQL database backend.
- Removes languages removed in upstream git repository.
- Improved quality checks processing.
- Added new checks (BB code, XML markup and newlines).
- Support for optional rebasing instead of merge.
- Possibility to relocate Weblate (for example to run it under /weblate path).
- Support for manually choosing file type in case autodetection fails.
- Better support for Android resources.
- Support for generating SSH key from web interface.
- More visible data exports.
- New buttons to enter some special characters.
- Support for exporting dictionary.
- Support for locking down whole Weblate installation.
- Checks for source strings and support for source strings review.
- Support for user comments for both translations and source strings.
- Better changes log tracking.
- Changes can now be monitored using RSS.
- Improved support for RTL languages.

4.25.8 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 textos contendo verificações com falha.
- Checking of requirements during setup.
- Documentation improvements.

4.25.9 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.

4.25.10 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.

4.26 Weblate 0.x series

4.26.1 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.
- Various code cleanups.
- Documentation improvements.
- Changed file layout.
- Update helper scripts to Django 1.4.
- Improved navigation while translating.
- Better handling of po file renames.
- Better validation while creating component.
- Integrated full setup into syncdb.
- Added list of recent changes to all translation pages.
- Check for not translated strings ignores format string only messages.

4.26.2 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.

4.26.3 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.

4.26.4 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.

4.26.5 Weblate 0.5

Released on February 12th 2012.

- **Support for machine translation using following online services:**
 - Apertium
 - Microsoft Translator
 - MyMemory
- Several new translations.
- Improved merging of upstream changes.
- Better handle concurrent git pull and translation.
- Propagating works for fuzzy changes as well.
- Propagating works also for file upload.
- Fixed file downloads while using FastCGI (and possibly others).

4.26.6 Weblate 0.4

Released on February 8th 2012.

- Added usage guide to documentation.
- Fixed API hooks not to require CSRF protection.

4.26.7 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.

4.26.8 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.

4.26.9 Weblate 0.1

Released on February 6th 2012.

- Initial release.

W

`wlc`, [135](#)
`wlc.config`, [135](#)
`wlc.main`, [136](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component)/(string:language)/, 108
ANY /, 86	GET /api/components/(string:project)/(string:component)/(string:language)/, 110
/api	GET /api/components/(string:project)/(string:component)/(string:language)/, 110
GET /api/, 89	GET /api/components/(string:project)/(string:component)/(string:language)/, 109
/api/addons	GET /api/components/(string:project)/(string:component)/(string:language)/, 108
GET /api/addons/, 123	GET /api/components/(string:project)/(string:component)/(string:language)/, 112
GET /api/addons/(int:id)/, 123	GET /api/components/(string:project)/(string:component)/(string:language)/, 110
PUT /api/addons/(int:id)/, 124	POST /api/components/(string:project)/(string:component)/(string:language)/, 123
DELETE /api/addons/(int:id)/, 124	POST /api/components/(string:project)/(string:component)/(string:language)/, 112
PATCH /api/addons/(int:id)/, 124	POST /api/components/(string:project)/(string:component)/(string:language)/, 108
/api/changes	POST /api/components/(string:project)/(string:component)/(string:language)/, 111
GET /api/changes/, 120	PUT /api/components/(string:project)/(string:component)/(string:language)/, 107
GET /api/changes/(int:id)/, 121	DELETE /api/components/(string:project)/(string:component)/(string:language)/, 108
/api/component-lists	DELETE /api/components/(string:project)/(string:component)/(string:language)/, 113
GET /api/component-lists/, 124	POST /api/components/(string:project)/(string:component)/(string:language)/, 106
GET /api/component-lists/(str:slug)/, 124	/api/groups
POST /api/component-lists/(str:slug)/components/, 125	GET /api/groups/, 92
PUT /api/component-lists/(str:slug)/, 125	GET /api/groups/(int:id)/, 92
DELETE /api/component-lists/(str:slug)/, 125	POST /api/groups/, 92
DELETE /api/component-lists/(str:slug)/components/(string:slug)/, 125	POST /api/groups/(int:id)/componentlists/, 95
PATCH /api/component-lists/(str:slug)/, 125	POST /api/groups/(int:id)/changes/, 94
/api/components	POST /api/groups/(int:id)/links/, 94
GET /api/components/, 104	
GET /api/components/(string:project)/(string:component)/(string:language)/, 104	
GET /api/components/(string:project)/(string:component)/(string:language)/, 108	
GET /api/components/(string:project)/(string:component)/(string:language)/, 112	

POST /api/groups/(int:id)/projects/, 94
 POST /api/groups/(int:id)/roles/, 94
 PUT /api/groups/(int:id)/, 93
 DELETE /api/groups/(int:id)/, 94
 DELETE /api/groups/(int:id)/componentlist/, 95
 DELETE /api/groups/(int:id)/components/(int:component_id)/, 94
 DELETE /api/groups/(int:id)/languages/(string:language_code)/, 94
 DELETE /api/groups/(int:id)/projects/(int:project_id)/, 94
 PATCH /api/groups/(int:id)/, 93

/api/languages

GET /api/languages/, 96
 GET /api/languages/(string:language)/, 96
 GET /api/languages/(string:language)/statistics/, 97
 POST /api/languages/, 96
 PUT /api/languages/(string:language)/, 97
 DELETE /api/languages/(string:language)/, 97
 PATCH /api/languages/(string:language)/, 97

/api/projects

GET /api/projects/, 98
 GET /api/projects/(string:project)/, 98
 GET /api/projects/(string:project)/changes/, 99
 GET /api/projects/(string:project)/components/, 100
 GET /api/projects/(string:project)/languages/, 103
 GET /api/projects/(string:project)/repository/, 99
 GET /api/projects/(string:project)/statistics/, 103
 POST /api/projects/, 98
 POST /api/projects/(string:project)/components/, 101
 POST /api/projects/(string:project)/repository/, 100
 PUT /api/projects/(string:project)/, 99
 DELETE /api/projects/(string:project)/, 99
 PATCH /api/projects/(string:project)/, 99

/api/roles

GET /api/roles/, 95
 GET /api/roles/(int:id)/, 95

POST /api/roles/, 95
 PUT /api/roles/(int:id)/, 95
 DELETE /api/roles/(int:id)/, 96
 PATCH /api/roles/(int:id)/, 96

/api/screenshots

GET /api/screenshots/, 121
 GET /api/screenshots/(int:id)/, 121
 GET /api/screenshots/(int:id)/file/(string:language_code)/, 121
 POST /api/screenshots/, 122
 POST /api/screenshots/(int:id)/file/, 121
 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)/, 122

/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)/, 116
 GET /api/translations/(string:project)/(string:component)/, 117
 GET /api/translations/(string:project)/(string:component)/, 118
 GET /api/translations/(string:project)/(string:component)/, 115
 POST /api/translations/(string:project)/(string:component)/, 116
 POST /api/translations/(string:project)/(string:component)/, 117
 POST /api/translations/(string:project)/(string:component)/, 117
 POST /api/translations/(string:project)/(string:component)/, 116
 DELETE /api/translations/(string:project)/(string:component)/, 115

/api/units

GET /api/units/, 118
 GET /api/units/(int:id)/, 118
 PUT /api/units/(int:id)/, 120
 DELETE /api/units/(int:id)/, 120
 PATCH /api/units/(int:id)/, 119

/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)/, 90
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)/,
  128

```

/hooks

```

GET /hooks/update/(string:project)/,
  126
GET /hooks/update/(string:project)/(string:component)/,
  126
POST /hooks/azure/, 127
POST /hooks/bitbucket/, 127
POST /hooks/gitea/, 127
POST /hooks/gitee/, 128
POST /hooks/github/, 126
POST /hooks/gitlab/, 127
POST /hooks/pagure/, 127

```


Símbolos

```
.XML resource file
  file format, 73
--add
  auto_translate opção de linha de comando, 338
--addon ADDON
  install_addon opção de linha de comando, 344
--age HOURS
  commit_pending opção de linha de comando, 339
--author USER@EXAMPLE.COM
  add_suggestions opção de linha de comando, 338
--base-file-template TEMPLATE
  import_project opção de linha de comando, 342
--check
  importusers opção de linha de comando, 344
--config PATH
  wlc opção de linha de comando, 131
--config-section SECTION
  wlc opção de linha de comando, 131
--configuration CONFIG
  install_addon opção de linha de comando, 344
--convert
  wlc opção de linha de comando, 133
--email USER@EXAMPLE.COM
  createadmin opção de linha de comando, 340
--file-format FORMAT
  import_project opção de linha de comando, 342
--force
  loadpo opção de linha de comando, 345
--force-commit
  pushgit opção de linha de comando, 346
--format {csv,json,text,html}
  wlc opção de linha de comando, 131
--ignore
  import_json opção de linha de comando, 341
--inconsistent
  auto_translate opção de linha de comando, 338
--input
  wlc opção de linha de comando, 133
--key KEY
  wlc opção de linha de comando, 131
--lang LANGUAGE
  loadpo opção de linha de comando, 345
--language-code
  list_translators opção de linha de comando, 345
--language-map LANGMAP
  import_memory opção de linha de comando, 341
--language-regex REGEX
  import_project opção de linha de comando, 342
--license NAME
  import_project opção de linha de comando, 342
--license-url URL
  import_project opção de linha de comando, 342
--main-component
  import_project opção de linha de comando, 342
--main-component COMPONENT
  import_json opção de linha de comando, 341
--mode MODE
  auto_translate opção de linha de comando, 338
--mt MT
  auto_translate opção de linha de comando, 338
--name
  createadmin opção de linha de comando, 340
```

```
--name-template TEMPLATE
  import_project opção de linha de
    comando, 342
--new-base-template TEMPLATE
  import_project opção de linha de
    comando, 342
--no-password
  createadmin opção de linha de co-
    mando, 340
--no-privs-update
  setupgroups opção de linha de co-
    mando, 346
--no-projects-update
  setupgroups opção de linha de co-
    mando, 346
--no-update
  setuplang opção de linha de co-
    mando, 347
--output
  wlc opção de linha de comando, 133
--overwrite
  auto_translate opção de linha de
    comando, 338
  wlc opção de linha de comando, 133
--password PASSWORD
  createadmin opção de linha de co-
    mando, 340
--project PROJECT
  import_json opção de linha de co-
    mando, 341
--source PROJECT/COMPONENT
  auto_translate opção de linha de
    comando, 338
--threshold THRESHOLD
  auto_translate opção de linha de
    comando, 338
--update
  createadmin opção de linha de co-
    mando, 340
  import_json opção de linha de co-
    mando, 341
  install_addon opção de linha de co-
    mando, 344
--url URL
  wlc opção de linha de comando, 131
--user USERNAME
  auto_translate opção de linha de
    comando, 338
--username USERNAME
  createadmin opção de linha de co-
    mando, 340
--vcs NAME
  import_project opção de linha de
    comando, 342
```

A

add_suggestions
weblate admin command, 338

```
add_suggestions opção de linha de co-
  mando
  --author USER@EXAMPLE.COM, 338
ADMINS
  setting, 180
AKISMET_API_KEY
  setting, 293
ALLOWED_HOSTS
  setting, 180
Android
  file format, 68
ANONYMOUS_USER_NAME
  setting, 293
API, 86, 130, 135
Apple strings
  file format, 68
ARB
  file format, 71
AUDITLOG_EXPIRY
  setting, 294
AUTH_LOCK_ATTEMPTS
  setting, 294
AUTH_TOKEN_VALID
  setting, 295
auto_translate
  weblate admin command, 338
auto_translate opção de linha de co-
  mando
  --add, 338
  --inconsistent, 338
  --mode MODE, 338
  --mt MT, 338
  --overwrite, 338
  --source PROJECT/COMPONENT, 338
  --threshold THRESHOLD, 338
  --user USERNAME, 338
```

```
AUTO_UPDATE
  setting, 294
```

```
AUTOFIX_LIST
  setting, 295
```

```
AVATAR_URL_PREFIX
  setting, 294
```

B

```
BACKGROUND_TASKS
  setting, 296
```

```
BASE_DIR
  setting, 296
```

BaseAddon (*classe em weblate.addons.base*), 381

```
BASIC_LANGUAGES
  setting, 296
```

```
bilingual
  translation, 59
```

C

```
can_install() (método de classe we-
  blate.addons.base.BaseAddon), 381
CELERY_BACKUP_OPTIONS, 141, 155
```

CELERY_BEAT_OPTIONS, 141, 155
CELERY_MAIN_OPTIONS, 141, 155
CELERY_MEMORY_OPTIONS, 141, 155
CELERY_NOTIFY_OPTIONS, 141, 155
celery_queues
 weblate admin command, 339
CELERY_TRANSLATE_OPTIONS, 141, 155
changes
 wlc opção de linha de comando, 132
CHECK_LIST
 setting, 297
checkgit
 weblate admin command, 339
cleanup
 wlc opção de linha de comando, 132
cleanuptrans
 weblate admin command, 339
Comma separated values
 file format, 73
Command (*classe em wlc.main*), 136
COMMENT_CLEANUP_DAYS
 setting, 297
commit
 wlc opção de linha de comando, 132
commit_pending
 weblate admin command, 339
commit_pending opção de linha de comando
 --age HOURS, 339
COMMIT_PENDING_HOURS
 setting, 298
commitgit
 weblate admin command, 339
configure() (método *weblate.addons.base.BaseAddon*), 381
CONTACT_FORM
 setting, 298
createadmin
 weblate admin command, 340
createadmin opção de linha de comando
 --email USER@EXAMPLE.COM, 340
 --name, 340
 --no-password, 340
 --password PASSWORD, 340
 --update, 340
 --username USERNAME, 340
CSP_CONNECT_SRC
 setting, 296
CSP_FONT_SRC
 setting, 296
CSP_IMG_SRC
 setting, 296
CSP_SCRIPT_SRC
 setting, 296
CSP_STYLE_SRC
 setting, 296
CSV
 file format, 73

D

daily() (método *weblate.addons.base.BaseAddon*), 381
DATA_DIR
 setting, 298
DATABASE_BACKUP
 setting, 298
DATABASES
 setting, 181
DEBUG
 setting, 181
DEFAULT_ACCESS_CONTROL
 setting, 299
DEFAULT_ADD_MESSAGE
 setting, 299
DEFAULT_ADDON_MESSAGE
 setting, 299
DEFAULT_ADDONS
 setting, 300
DEFAULT_AUTO_WATCH
 setting, 299
DEFAULT_COMMIT_MESSAGE
 setting, 299
DEFAULT_COMMITTER_EMAIL
 setting, 300
DEFAULT_COMMITTER_NAME
 setting, 300
DEFAULT_DELETE_MESSAGE
 setting, 299
DEFAULT_FROM_EMAIL
 setting, 181
DEFAULT_LANGUAGE
 setting, 300
DEFAULT_MERGE_MESSAGE
 setting, 299
DEFAULT_MERGE_STYLE
 setting, 301
DEFAULT_PULL_MESSAGE
 setting, 301
DEFAULT_RESTRICTED_COMPONENT
 setting, 299
DEFAULT_SHARED_TM
 setting, 301
DEFAULT_TRANSLATION_PROPAGATION
 setting, 301
download
 wlc opção de linha de comando, 133
DTD
 file format, 75
dump_memory
 weblate admin command, 340
dumpuserdata
 weblate admin command, 340

E

ENABLE_AVATARS
 setting, 301
ENABLE_HOOKS

setting, 301
 ENABLE_HTTPS
 setting, 302
 ENABLE_SHARING
 setting, 302

F

file format
 .XML resource file, 73
 Android, 68
 Apple strings, 68
 ARB, 71
 Comma separated values, 73
 CSV, 73
 DTD, 75
 gettext, 62
 go-i18n, 71
 GWT properties, 66
 i18next, 70
 INI translations, 66
 Java properties, 65
 Joomla translations, 67
 JSON, 69
 PHP strings, 69
 PO, 62
 Qt, 67
 RC, 75
 RESX, 73
 Ruby YAML, 74
 Ruby YAML Ain't Markup Language, 74
 string resources, 68
 TS, 67
 XLIFF, 63
 XML, 75
 YAML, 74
 YAML Ain't Markup Language, 74

G

get() (método wlc.Weblate), 135
 get_add_form() (método de classe weblate.addons.base.BaseAddon), 381
 GET_HELP_URL
 setting, 302
 get_settings_form() (método weblate.addons.base.BaseAddon), 381
 gettext
 file format, 62
 GITHUB_CREDENTIALS
 setting, 303
 GITHUB_TOKEN
 setting, 303
 GITHUB_USERNAME
 setting, 303
 GITLAB_CREDENTIALS
 setting, 302
 GITLAB_TOKEN
 setting, 303
 GITLAB_USERNAME

setting, 302
 go-i18n
 file format, 71
 GOOGLE_ANALYTICS_ID
 setting, 303
 GWT properties
 file format, 66

H

HIDE_REPO_CREDENTIALS
 setting, 304
 HIDE_VERSION
 setting, 304

I

i18next
 file format, 70
 import_demo
 weblate admin command, 340
 import_json
 weblate admin command, 341
 import_json opção de linha de comando
 --ignore, 341
 --main-component COMPONENT, 341
 --project PROJECT, 341
 --update, 341
 import_memory
 weblate admin command, 341
 import_memory opção de linha de comando
 --language-map LANGMAP, 341
 import_project
 weblate admin command, 342
 import_project opção de linha de comando
 --base-file-template TEMPLATE, 342
 --file-format FORMAT, 342
 --language-regex REGEX, 342
 --license NAME, 342
 --license-url URL, 342
 --main-component, 342
 --name-template TEMPLATE, 342
 --new-base-template TEMPLATE, 342
 --vcs NAME, 342
 importuserdata
 weblate admin command, 344
 importusers
 weblate admin command, 344
 importusers opção de linha de comando
 --check, 344
 INI translations
 file format, 66
 install_addon
 weblate admin command, 344
 install_addon opção de linha de comando
 --addon ADDON, 344
 --configuration CONFIG, 344

- update, 344
- IP_BEHIND_REVERSE_PROXY
 - setting, 304
- IP_PROXY_HEADER
 - setting, 304
- IP_PROXY_OFFSET
 - setting, 305
- iPad
 - translation, 68
- iPhone
 - translation, 68
- J**
- Java properties
 - file format, 65
- Joomla translations
 - file format, 67
- JSON
 - file format, 69
- L**
- LEGAL_URL
 - setting, 305
- LICENSE_EXTRA
 - setting, 305
- LICENSE_FILTER
 - setting, 306
- LICENSE_REQUIRED
 - setting, 306
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
 - setting, 306
- list_languages
 - weblate admin command, 344
- list_translators
 - weblate admin command, 345
- list_translators opção de linha de comando
 - language-code, 345
- list_versions
 - weblate admin command, 345
- list-components
 - wlc opção de linha de comando, 132
- list-languages
 - wlc opção de linha de comando, 132
- list-projects
 - wlc opção de linha de comando, 132
- list-translations
 - wlc opção de linha de comando, 132
- load() (*método wlc.config.WeblateConfig*), 135
- loadpo
 - weblate admin command, 345
- loadpo opção de linha de comando
 - force, 345
 - lang LANGUAGE, 345
- LOCALIZE_CDN_PATH
 - setting, 306
- LOCALIZE_CDN_URL
 - setting, 306

- lock
 - wlc opção de linha de comando, 132
- lock_translation
 - weblate admin command, 345
- lock-status
 - wlc opção de linha de comando, 132
- LOGIN_REQUIRED_URLS
 - setting, 307
- LOGIN_REQUIRED_URLS_EXCEPTIONS
 - setting, 307
- ls
 - wlc opção de linha de comando, 132

M

- MACHINE_TRANSLATION_SERVICES
 - setting, 308
- main() (*no módulo wlc.main*), 136
- MATOMO_SITE_ID
 - setting, 307
- MATOMO_URL
 - setting, 308
- monolingual
 - translation, 59
- move_language
 - weblate admin command, 346
- MT_APERTIUM_APY
 - setting, 308
- MT_AWS_ACCESS_KEY_ID
 - setting, 309
- MT_AWS_REGION
 - setting, 309
- MT_AWS_SECRET_ACCESS_KEY
 - setting, 309
- MT_BAIDU_ID
 - setting, 309
- MT_BAIDU_SECRET
 - setting, 309
- MT_DEEPL_API_VERSION
 - setting, 309
- MT_DEEPL_KEY
 - setting, 310
- MT_GOOGLE_CREDENTIALS
 - setting, 310
- MT_GOOGLE_KEY
 - setting, 310
- MT_GOOGLE_LOCATION
 - setting, 310
- MT_GOOGLE_PROJECT
 - setting, 310
- MT_MICROSOFT_BASE_URL
 - setting, 310
- MT_MICROSOFT_COGNITIVE_KEY
 - setting, 311
- MT_MICROSOFT_ENDPOINT_URL
 - setting, 311
- MT_MICROSOFT_REGION
 - setting, 311
- MT_MODERNMT_KEY

setting, 311
 MT_MODERNMT_URL
 setting, 311
 MT_MYMEMORY_EMAIL
 setting, 311
 MT_MYMEMORY_KEY
 setting, 312
 MT_MYMEMORY_USER
 setting, 312
 MT_NETEASE_KEY
 setting, 312
 MT_NETEASE_SECRET
 setting, 312
 MT_SAP_BASE_URL
 setting, 313
 MT_SAP_PASSWORD
 setting, 313
 MT_SAP_SANDBOX_APIKEY
 setting, 313
 MT_SAP_USE_MT
 setting, 313
 MT_SAP_USERNAME
 setting, 313
 MT_SERVICES
 setting, 308
 MT_TMSERVER
 setting, 312
 MT_YANDEX_KEY
 setting, 312
 MT_YOUDAO_ID
 setting, 312
 MT_YOUDAO_SECRET
 setting, 313
 módulo
 wlc, 135
 wlc.config, 135
 wlc.main, 136

N

NEARBY_MESSAGES
 setting, 314

P

PAGURE_CREDENTIALS
 setting, 314
 PAGURE_TOKEN
 setting, 314
 PAGURE_USERNAME
 setting, 314
 PHP strings
 file format, 69
 PIWIK_SITE_ID
 setting, 307
 PIWIK_URL
 setting, 308
 PO
 file format, 62
 post() (método wlc.Weblate), 135

post_add() (método
 blate.addons.base.BaseAddon), 381
 post_commit() (método
 blate.addons.base.BaseAddon), 381
 post_push() (método
 blate.addons.base.BaseAddon), 381
 post_update() (método
 blate.addons.base.BaseAddon), 381
 pre_commit() (método
 blate.addons.base.BaseAddon), 381
 pre_push() (método
 blate.addons.base.BaseAddon), 381
 pre_update() (método
 blate.addons.base.BaseAddon), 381
 pull
 wlc opção de linha de comando, 132
 push
 wlc opção de linha de comando, 132
 pushgit
 weblate admin command, 346
 pushgit opção de linha de comando
 --force-commit, 346
 Python, 135

Q

Qt
 file format, 67

R

RATELIMIT_ATTEMPTS
 setting, 314
 RATELIMIT_LOCKOUT
 setting, 315
 RATELIMIT_WINDOW
 setting, 315
 RC
 file format, 75
 register_command() (no módulo wlc.main), 136
 REGISTRATION_ALLOW_BACKENDS
 setting, 315
 REGISTRATION_CAPTCHA
 setting, 315
 REGISTRATION_EMAIL_MATCH
 setting, 316
 REGISTRATION_OPEN
 setting, 316
 repo
 wlc opção de linha de comando, 132
 REPOSITORY_ALERT_THRESHOLD
 setting, 316
 REQUIRE_LOGIN
 setting, 316
 reset
 wlc opção de linha de comando, 132
 REST, 86
 RESX
 file format, 73
 RFC

RFC 4646, 59
Ruby YAML
 file format, 74
Ruby YAML Ain't Markup Language
 file format, 74

S

`save_state()` (método *blate.addons.base.BaseAddon*), 381
`SECRET_KEY`
 setting, 181
`SENTRY_DSN`
 setting, 317
`SERVER_EMAIL`
 setting, 181
`SESSION_COOKIE_AGE_AUTHENTICATED`
 setting, 317
`SESSION_ENGINE`
 setting, 180
`setting`
 ADMINS, 180
 AKISMET_API_KEY, 293
 ALLOWED_HOSTS, 180
 ANONYMOUS_USER_NAME, 293
 AUDITLOG_EXPIRY, 294
 AUTH_LOCK_ATTEMPTS, 294
 AUTH_TOKEN_VALID, 295
 AUTO_UPDATE, 294
 AUTOFIX_LIST, 295
 AVATAR_URL_PREFIX, 294
 BACKGROUND_TASKS, 296
 BASE_DIR, 296
 BASIC_LANGUAGES, 296
 CHECK_LIST, 297
 COMMENT_CLEANUP_DAYS, 297
 COMMIT_PENDING_HOURS, 298
 CONTACT_FORM, 298
 CSP_CONNECT_SRC, 296
 CSP_FONT_SRC, 296
 CSP_IMG_SRC, 296
 CSP_SCRIPT_SRC, 296
 CSP_STYLE_SRC, 296
 DATA_DIR, 298
 DATABASE_BACKUP, 298
 DATABASES, 181
 DEBUG, 181
 DEFAULT_ACCESS_CONTROL, 299
 DEFAULT_ADD_MESSAGE, 299
 DEFAULT_ADDON_MESSAGE, 299
 DEFAULT_ADDONS, 300
 DEFAULT_AUTO_WATCH, 299
 DEFAULT_COMMIT_MESSAGE, 299
 DEFAULT_COMMITER_EMAIL, 300
 DEFAULT_COMMITER_NAME, 300
 DEFAULT_DELETE_MESSAGE, 299
 DEFAULT_FROM_EMAIL, 181
 DEFAULT_LANGUAGE, 300
 DEFAULT_MERGE_MESSAGE, 299

we-

 DEFAULT_MERGE_STYLE, 301
 DEFAULT_PULL_MESSAGE, 301
 DEFAULT_RESTRICTED_COMPONENT, 299
 DEFAULT_SHARED_TM, 301
 DEFAULT_TRANSLATION_PROPAGATION,
 301
 ENABLE_AVATARS, 301
 ENABLE_HOOKS, 301
 ENABLE_HTTPS, 302
 ENABLE_SHARING, 302
 GET_HELP_URL, 302
 GITHUB_CREDENTIALS, 303
 GITHUB_TOKEN, 303
 GITHUB_USERNAME, 303
 GITLAB_CREDENTIALS, 302
 GITLAB_TOKEN, 303
 GITLAB_USERNAME, 302
 GOOGLE_ANALYTICS_ID, 303
 HIDE_REPO_CREDENTIALS, 304
 HIDE_VERSION, 304
 IP_BEHIND_REVERSE_PROXY, 304
 IP_PROXY_HEADER, 304
 IP_PROXY_OFFSET, 305
 LEGAL_URL, 305
 LICENSE_EXTRA, 305
 LICENSE_FILTER, 306
 LICENSE_REQUIRED, 306
 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH,
 306
 LOCALIZE_CDN_PATH, 306
 LOCALIZE_CDN_URL, 306
 LOGIN_REQUIRED_URLS, 307
 LOGIN_REQUIRED_URLS_EXCEPTIONS, 307
 MACHINE_TRANSLATION_SERVICES, 308
 MATOMO_SITE_ID, 307
 MATOMO_URL, 308
 MT_APERTIUM_API_KEY, 308
 MT_AWS_ACCESS_KEY_ID, 309
 MT_AWS_REGION, 309
 MT_AWS_SECRET_ACCESS_KEY, 309
 MT_BAIDU_ID, 309
 MT_BAIDU_SECRET, 309
 MT_DEEPL_API_VERSION, 309
 MT_DEEPL_KEY, 310
 MT_GOOGLE_CREDENTIALS, 310
 MT_GOOGLE_KEY, 310
 MT_GOOGLE_LOCATION, 310
 MT_GOOGLE_PROJECT, 310
 MT_MICROSOFT_BASE_URL, 310
 MT_MICROSOFT_COGNITIVE_KEY, 311
 MT_MICROSOFT_ENDPOINT_URL, 311
 MT_MICROSOFT_REGION, 311
 MT_MODERNMT_KEY, 311
 MT_MODERNMT_URL, 311
 MT_MYMEMORY_EMAIL, 311
 MT_MYMEMORY_KEY, 312
 MT_MYMEMORY_USER, 312
 MT_NETEASE_KEY, 312

- MT_NETEASE_SECRET, 312
- MT_SAP_BASE_URL, 313
- MT_SAP_PASSWORD, 313
- MT_SAP_SANDBOX_APIKEY, 313
- MT_SAP_USE_MT, 313
- MT_SAP_USERNAME, 313
- MT_SERVICES, 308
- MT_TMSERVER, 312
- MT_YANDEX_KEY, 312
- MT_YOUDAO_ID, 312
- MT_YOUDAO_SECRET, 313
- NEARBY_MESSAGES, 314
- PAGURE_CREDENTIALS, 314
- PAGURE_TOKEN, 314
- PAGURE_USERNAME, 314
- PIWIK_SITE_ID, 307
- PIWIK_URL, 308
- RATELIMIT_ATTEMPTS, 314
- RATELIMIT_LOCKOUT, 315
- RATELIMIT_WINDOW, 315
- REGISTRATION_ALLOW_BACKENDS, 315
- REGISTRATION_CAPTCHA, 315
- REGISTRATION_EMAIL_MATCH, 316
- REGISTRATION_OPEN, 316
- REPOSITORY_ALERT_THRESHOLD, 316
- REQUIRE_LOGIN, 316
- SECRET_KEY, 181
- SENTRY_DSN, 317
- SERVER_EMAIL, 181
- SESSION_COOKIE_AGE_AUTHENTICATED, 317
- SESSION_ENGINE, 180
- SIMPLIFY_LANGUAGES, 317
- SINGLE_PROJECT, 318
- SITE_DOMAIN, 317
- SITE_TITLE, 318
- SPECIAL_CHARS, 318
- STATUS_URL, 318
- SUGGESTION_CLEANUP_DAYS, 318
- UPDATE_LANGUAGES, 318
- URL_PREFIX, 319
- VCS_BACKENDS, 319
- VCS_CLONE_DEPTH, 319
- WEBLATE_ADDONS, 320
- WEBLATE_EXPORTERS, 321
- WEBLATE_FORMATS, 321
- WEBLATE_GPG_IDENTITY, 321
- WEBSITE_REQUIRED, 321
- setupgroups
 - weblate admin command, 346
- setupgroups opção de linha de comando
 - no-privs-update, 346
 - no-projects-update, 346
- setuplang
 - weblate admin command, 347
- setuplang opção de linha de comando
 - no-update, 347
- show
 - wlc opção de linha de comando, 132
- SIMPLIFY_LANGUAGES
 - setting, 317
- SINGLE_PROJECT
 - setting, 318
- SITE_DOMAIN
 - setting, 317
- SITE_TITLE
 - setting, 318
- SPECIAL_CHARS
 - setting, 318
- statistics
 - wlc opção de linha de comando, 132
- STATUS_URL
 - setting, 318
- stay_on_create (atributo *weblate.addons.base.BaseAddon*), 381
- store_post_load() (método *weblate.addons.base.BaseAddon*), 382
- string resources
 - file format, 68
- SUGGESTION_CLEANUP_DAYS
 - setting, 318
- T
 - translation
 - bilingual, 59
 - iPad, 68
 - iPhone, 68
 - monolingual, 59
 - TS
 - file format, 67
- U
 - unit_pre_create() (método *weblate.addons.base.BaseAddon*), 382
 - unlock
 - wlc opção de linha de comando, 132
 - unlock_translation
 - weblate admin command, 346
 - UPDATE_LANGUAGES
 - setting, 318
 - updatechecks
 - weblate admin command, 347
 - updategit
 - weblate admin command, 347
 - upload
 - wlc opção de linha de comando, 133
 - URL_PREFIX
 - setting, 319
 - UWSGI_WORKERS, 141, 155
- V
 - VCS_BACKENDS
 - setting, 319
 - VCS_CLONE_DEPTH
 - setting, 319
 - version

- wlc opção de linha de comando, 132
- váriavel de ambiente
 - CELERY_BACKUP_OPTIONS, 141, 155
 - CELERY_BEAT_OPTIONS, 141, 155
 - CELERY_MAIN_OPTIONS, 141, 155
 - CELERY_MEMORY_OPTIONS, 141, 155
 - CELERY_NOTIFY_OPTIONS, 141, 155
 - CELERY_TRANSLATE_OPTIONS, 141, 155
 - POSTGRES_ALTER_ROLE, 152
 - POSTGRES_DATABASE, 151
 - POSTGRES_HOST, 151
 - POSTGRES_PASSWORD, 151
 - POSTGRES_PORT, 151
 - POSTGRES_SSL_MODE, 151
 - POSTGRES_USER, 151
 - REDIS_DB, 152
 - REDIS_HOST, 152
 - REDIS_PASSWORD, 152
 - REDIS_PORT, 152
 - REDIS_TLS, 152
 - REDIS_VERIFY_SSL, 152
 - ROLLBAR_ENVIRONMENT, 154
 - ROLLBAR_KEY, 154
 - SENTRY_DSN, 154
 - SENTRY_ENVIRONMENT, 154
 - SOCIAL_AUTH_SLACK_SECRET, 151
 - UWSGI_WORKERS, 141, 155, 156
 - WEBLATE_ADD_ADDONS, 155
 - WEBLATE_ADD_APPS, 155
 - WEBLATE_ADD_AUTOFIX, 155
 - WEBLATE_ADD_CHECK, 155
 - WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 144
 - WEBLATE_ADMIN_EMAIL, 141--143, 147
 - WEBLATE_ADMIN_NAME, 141--143
 - WEBLATE_ADMIN_PASSWORD, 138, 141--143
 - WEBLATE_AKISMET_API_KEY, 146, 354
 - WEBLATE_ALLOWED_HOSTS, 143, 180, 185, 318
 - WEBLATE_AUTH_LDAP_BIND_DN, 148
 - WEBLATE_AUTH_LDAP_BIND_PASSWORD, 148
 - WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERENCE, 148
 - WEBLATE_AUTH_LDAP_SERVER_URI, 148
 - WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 148
 - WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 148
 - WEBLATE_AUTH_LDAP_USER_SEARCH, 148
 - WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 148
 - WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 148
 - WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 148
 - WEBLATE_BASIC_LANGUAGES, 146
 - WEBLATE_CONTACT_FORM, 143
 - WEBLATE_CSP_CONNECT_SRC, 146
 - WEBLATE_CSP_FONT_SRC, 146
 - WEBLATE_CSP_IMG_SRC, 146
 - WEBLATE_CSP_SCRIPT_SRC, 146
 - WEBLATE_CSP_STYLE_SRC, 146
 - WEBLATE_DATABASE_BACKUP, 152
 - WEBLATE_DEBUG, 142
 - WEBLATE_DEFAULT_ACCESS_CONTROL, 145
 - WEBLATE_DEFAULT_AUTO_WATCH, 146
 - WEBLATE_DEFAULT_COMMITTER_EMAIL, 145
 - WEBLATE_DEFAULT_COMMITTER_NAME, 146
 - WEBLATE_DEFAULT_FROM_EMAIL, 143
 - WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 145
 - WEBLATE_DEFAULT_SHARED_TM, 146
 - WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 145
 - WEBLATE_EMAIL_BACKEND, 153
 - WEBLATE_EMAIL_HOST, 153
 - WEBLATE_EMAIL_HOST_PASSWORD, 153
 - WEBLATE_EMAIL_HOST_USER, 153
 - WEBLATE_EMAIL_PORT, 153
 - WEBLATE_EMAIL_USE_SSL, 153
 - WEBLATE_EMAIL_USE_TLS, 153
 - WEBLATE_ENABLE_AVATARS, 147
 - WEBLATE_ENABLE_HTTPS, 143
 - WEBLATE_GET_HELP_URL, 154
 - WEBLATE_GITHUB_TOKEN, 145
 - WEBLATE_GITHUB_USERNAME, 145
 - WEBLATE_GITLAB_TOKEN, 145
 - WEBLATE_GITLAB_USERNAME, 145
 - WEBLATE_GOOGLE_ANALYTICS_ID, 145
 - WEBLATE_GPG_IDENTITY, 146
 - WEBLATE_HIDE_VERSION, 146
 - WEBLATE_IP_PROXY_HEADER, 144
 - WEBLATE_LEGAL_URL, 154
 - WEBLATE_LICENSE_FILTER, 146
 - WEBLATE_LICENSE_REQUIRED, 146
 - WEBLATE_LOCALIZE_CDN_PATH, 154
 - WEBLATE_LOCALIZE_CDN_URL, 154
 - WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 144
 - WEBLATE_LOGLEVEL, 142
 - WEBLATE_MT_APERTIUM_API, 147
 - WEBLATE_MT_AWS_ACCESS_KEY_ID, 147
 - WEBLATE_MT_AWS_REGION, 147
 - WEBLATE_MT_AWS_SECRET_ACCESS_KEY, 147
 - WEBLATE_MT_DEEPL_API_VERSION, 147
 - WEBLATE_MT_DEEPL_KEY, 147
 - WEBLATE_MT_GLOSBE_ENABLED, 147
 - WEBLATE_MT_GOOGLE_KEY, 147
 - WEBLATE_MT_MICROSOFT_BASE_URL, 147
 - WEBLATE_MT_MICROSOFT_COGNITIVE_KEY, 147
 - WEBLATE_MT_MICROSOFT_ENDPOINT_URL, 147
 - WEBLATE_MT_MICROSOFT_REGION, 147

WEBLATE_MT_MICROSOFT_TERMINOLOGY_ENABLED, 149
 148
 WEBLATE_MT_MODERNMT_KEY, 147
 WEBLATE_MT_MYMEMORY_ENABLED, 147
 WEBLATE_MT_SAP_BASE_URL, 148
 WEBLATE_MT_SAP_PASSWORD, 148
 WEBLATE_MT_SAP_SANDBOX_APIKEY, 148
 WEBLATE_MT_SAP_USE_MT, 148
 WEBLATE_MT_SAP_USERNAME, 148
 WEBLATE_NO_EMAIL_AUTH, 151
 WEBLATE_PAGURE_TOKEN, 145
 WEBLATE_PAGURE_USERNAME, 145
 WEBLATE_RATELIMIT_ATTEMPTS, 146, 356
 WEBLATE_RATELIMIT_LOCKOUT, 146
 WEBLATE_RATELIMIT_WINDOW, 146
 WEBLATE_REGISTRATION_ALLOW_BACKENDS, 143
 WEBLATE_REGISTRATION_OPEN, 143
 WEBLATE_REMOVE_ADDONS, 155
 WEBLATE_REMOVE_APPS, 155
 WEBLATE_REMOVE_AUTOFIX, 155
 WEBLATE_REMOVE_CHECK, 155
 WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTED, 144
 WEBLATE_REQUIRE_LOGIN, 144, 317
 WEBLATE_SAML_IDP_ENTITY_ID, 151
 WEBLATE_SAML_IDP_URL, 151
 WEBLATE_SAML_IDP_X509CERT, 151
 WEBLATE_SECURE_PROXY_SSL_HEADER, 144
 WEBLATE_SERVER_EMAIL, 143
 WEBLATE_SERVICE, 141, 156
 WEBLATE_SILENCED_SYSTEM_CHECKS, 146, 208
 WEBLATE_SIMPLIFY_LANGUAGES, 145
 WEBLATE_SITE_DOMAIN, 142, 183, 201, 318
 WEBLATE_SITE_TITLE, 142
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 150
 WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 150
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 150
 WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET, 150
 WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 149
 WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 149
 WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 149
 WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 149
 WEBLATE_SOCIAL_AUTH_FEDORA, 151
 WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 149
 WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 149
 WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 150
 WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 150
 WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 150
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 150
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 150
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 150
 WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS, 150
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 150
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 150
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 150
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 150
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 150
 WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 150
 WEBLATE_SOCIAL_AUTH_OPENSUSE, 151
 WEBLATE_SOCIAL_AUTH_SLACK_KEY, 151
 WEBLATE_SOCIAL_AUTH_UBUNTU, 151
 WEBLATE_STATUS_URL, 154
 WEBLATE_TIME_ZONE, 143
 WEBLATE_URL_PREFIX, 146
 WEBLATE_WEBSITE_REQUIRED, 146
 WEBLATE_WORKERS, 141, 155, 156
 WL_BRANCH, 290
 WL_COMPONENT_NAME, 291
 WL_COMPONENT_SLUG, 290
 WL_COMPONENT_URL, 291
 WL_ENGAGE_URL, 291
 WL_FILE_FORMAT, 290
 WL_FILEMASK, 290
 WL_LANGUAGE, 290
 WL_NEW_BASE, 290
 WL_FETCH, 290
 WL_PREVIOUS_HEAD, 290
 WL_PROJECT_NAME, 291
 WL_PROJECT_SLUG, 291
 WL_REPO, 290
 WL_TEMPLATE, 290
 WL_VCS, 290

W

Weblate (*classe em wlc*), 135
 weblate admin command
 add_suggestions, 338
 auto_translate, 338
 celery_queues, 339
 checkgit, 339

- cleanuptrans, 339
- commit_pending, 339
- commitgit, 339
- createadmin, 340
- dump_memory, 340
- dumpuserdata, 340
- import_demo, 340
- import_json, 341
- import_memory, 341
- import_project, 342
- importuserdata, 344
- importusers, 344
- install_addon, 344
- list_languages, 344
- list_translators, 345
- list_versions, 345
- loadpo, 345
- lock_translation, 345
- move_language, 346
- pushgit, 346
- setupgroups, 346
- setuplang, 347
- unlock_translation, 346
- updatechecks, 347
- updategit, 347
- WEBLATE_ADDONS
 - setting, 320
- WEBLATE_ADMIN_EMAIL, 141--143, 147
- WEBLATE_ADMIN_NAME, 141--143
- WEBLATE_ADMIN_PASSWORD, 138, 141--143
- WEBLATE_AKISMET_API_KEY, 354
- WEBLATE_ALLOWED_HOSTS, 180, 185, 318
- WEBLATE_EMAIL_PORT, 153
- WEBLATE_EMAIL_USE_SSL, 153
- WEBLATE_EMAIL_USE_TLS, 153
- WEBLATE_EXPORTERS
 - setting, 321
- WEBLATE_FORMATS
 - setting, 321
- WEBLATE_GPG_IDENTITY
 - setting, 321
- WEBLATE_LOCALIZE_CDN_PATH, 154
- WEBLATE_RATELIMIT_ATTEMPTS, 356
- WEBLATE_REQUIRE_LOGIN, 317
- WEBLATE_SECURE_PROXY_SSL_HEADER, 144
- WEBLATE_SERVICE, 141
- WEBLATE_SILENCED_SYSTEM_CHECKS, 208
- WEBLATE_SITE_DOMAIN, 183, 201, 318
- WEBLATE_WORKERS, 141, 155, 156
- WeblateConfig (*classe em wlc.config*), 135
- WeblateException, 135
- WEBSITE_REQUIRED
 - setting, 321
- wlc, 130
 - módulo, 135
- wlc opção de linha de comando
 - config PATH, 131
 - config-section SECTION, 131

- convert, 133
- format {csv,json,text,html}, 131
- input, 133
- key KEY, 131
- output, 133
- overwrite, 133
- url URL, 131
- changes, 132
- cleanup, 132
- commit, 132
- download, 133
- list-components, 132
- list-languages, 132
- list-projects, 132
- list-translations, 132
- lock, 132
- lock-status, 132
- ls, 132
- pull, 132
- push, 132
- repo, 132
- reset, 132
- show, 132
- statistics, 132
- unlock, 132
- upload, 133
- version, 132
- wlc.config
 - módulo, 135
- wlc.main
 - módulo, 136

X

- XLIFF
 - file format, 63
- XML
 - file format, 75

Y

- YAML
 - file format, 74
- YAML Ain't Markup Language
 - file format, 74