The Weblate Manual

Versão 4.14.1

Michal Čihař

15 set., 2022

Documentação de utilizador

1 Documentação de utilizador **1**
1.1 Básico do Weblate . . . 1
1.2 Registo e perfil de utilizador . . . 1
1.3 Traduzir a usar o Weblate . . . 10
1.4 Descarregar e enviar traduções . . . 21
1.5 Glossário . . . 25
1.6 Verificações e correções . . . 27
1.7 Pesquisando . . . 60
1.8 Fluxos de trabalho de tradução . . . 66
1.9 Perguntas frequentes . . . 69
1.10 Formatos de ficheiros suportados . . . 78
1.11 Integração de controlo de versões . . . 99
1.12 API REST do Weblate . . . 107
1.13 Cliente Weblate . . . 155
1.14 API Python do Weblate . . . 160

2 Documentação de administrador **162**
2.1 Instruções de configuração . . . 162
2.2 Implantações de Weblate . . . 227
2.3 Atualizando o Weblate . . . 228
2.4 Fazer backup e mover o Weblate . . . 236
2.5 Autenticação . . . 243
2.6 Controlo de acesso . . . 253
2.7 Projetos de tradução . . . 262
2.8 Definições de idioma . . . 280
2.9 Tradução contínua . . . 283
2.10 Licenciar traduções . . . 293
2.11 Processo de tradução . . . 294
2.12 Verificações e correções . . . 300
2.13 Configuring automatic suggestions . . . 311
2.14 Extensões . . . 322
2.15 Memória de Tradução . . . 340
2.16 Configuração . . . 342
2.17 Amostra de configuração . . . 367
2.18 Comandos de gerência . . . 383
2.19 Anúncios . . . 395
2.20 Lista de componentes . . . 397
2.21 Módulos opcionais do Weblate . . . 398
2.22 Personalizar o Weblate . . . 403
2.23 Interface de gestão . . . 405
2.24 Obter suporte para o Weblate . . . 414

2.25 Documentos legais . . . 417

3 Documentação de colaborador **419**
3.1 Contribuir ao Weblate . . . 419
3.2 Comea çar a contribuir código para o Weblate . . . 421
3.3 Código-fonte do Weblate . . . 425
3.4 Depuração do Weblate . . . 426
3.5 Componentes internos do Weblate . . . 428
3.6 Desenvolver extensões . . . 429
3.7 Frontend do Weblate . . . 431
3.8 Relatando problemas no Weblate . . . 432
3.9 Conjunto de testes e integração contínua do Weblate . . . 433
3.10 Esquemas de dados . . . 435
3.11 Criando lançamento do Weblate . . . 438
3.12 Segurança e privacidade . . . 439
3.13 Contributing to Weblate modules . . . 440
3.14 Sobre o Weblate . . . 441
3.15 Licença . . . 442

4 Histórico de alterações **443**
4.1 Weblate 4.14.1 . . . 443
4.2 Weblate 4.14 . . . 444
4.3 Weblate 4.13.1 . . . 444
4.4 Weblate 4.13 . . . 444
4.5 Weblate 4.12.2 . . . 445
4.6 Weblate 4.12.1 . . . 445
4.7 Weblate 4.12 . . . 446
4.8 Weblate 4.11.2 . . . 446
4.9 Weblate 4.11.1 . . . 446
4.10 Weblate 4.11 . . . 447
4.11 Weblate 4.10.1 . . . 447
4.12 Weblate 4.10 . . . 448
4.13 Weblate 4.9.1 . . . 448
4.14 Weblate 4.9 . . . 448
4.15 Weblate 4.8.1 . . . 449
4.16 Weblate 4.8 . . . 450
4.17 Weblate 4.7.2 . . . 450
4.18 Weblate 4.7.1 . . . 450
4.19 Weblate 4.7 . . . 451
4.20 Weblate 4.6.2 . . . 451
4.21 Weblate 4.6.1 . . . 452
4.22 Weblate 4.6 . . . 452
4.23 Weblate 4.5.3 . . . 453
4.24 Weblate 4.5.2 . . . 453
4.25 Weblate 4.5.1 . . . 453
4.26 Weblate 4.5 . . . 454
4.27 Weblate 4.4.2 . . . 454
4.28 Weblate 4.4.1 . . . 455
4.29 Weblate 4.4 . . . 455
4.30 Weblate 4.3.2 . . . 456
4.31 Weblate 4.3.1 . . . 456
4.32 Weblate 4.3 . . . 457
4.33 Weblate 4.2.2 . . . 458
4.34 Weblate 4.2.1 . . . 458
4.35 Weblate 4.2 . . . 458
4.36 Weblate 4.1.1 . . . 459
4.37 Weblate 4.1 . . . 459
4.38 Weblate 4.0.4 . . . 460

4.39 Weblate 4.0.3 461
4.40 Weblate 4.0.2 461
4.41 Weblate 4.0.1 461
4.42 Weblate 4.0 462
4.43 Série 3.x do Weblate 462
4.44 Série 2.x do Weblate 474
4.45 Série 1.x do Weblate 485
4.46 Série 0.x do Weblate 489

Índice de Módulos do Python **493**

HTTP Routing Table **494**

Índice **497**

CAPÍTULO 1

Documentação de utilizador

1.1 Básico do Weblate

1.1.1 Estrutura de projetos e componentes

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

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

Veja também:

../devel/integration

1.2 Registo e perfil de utilizador

1.2.1 Registo

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

Pode registar-se seguindo alguns passos simples:

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

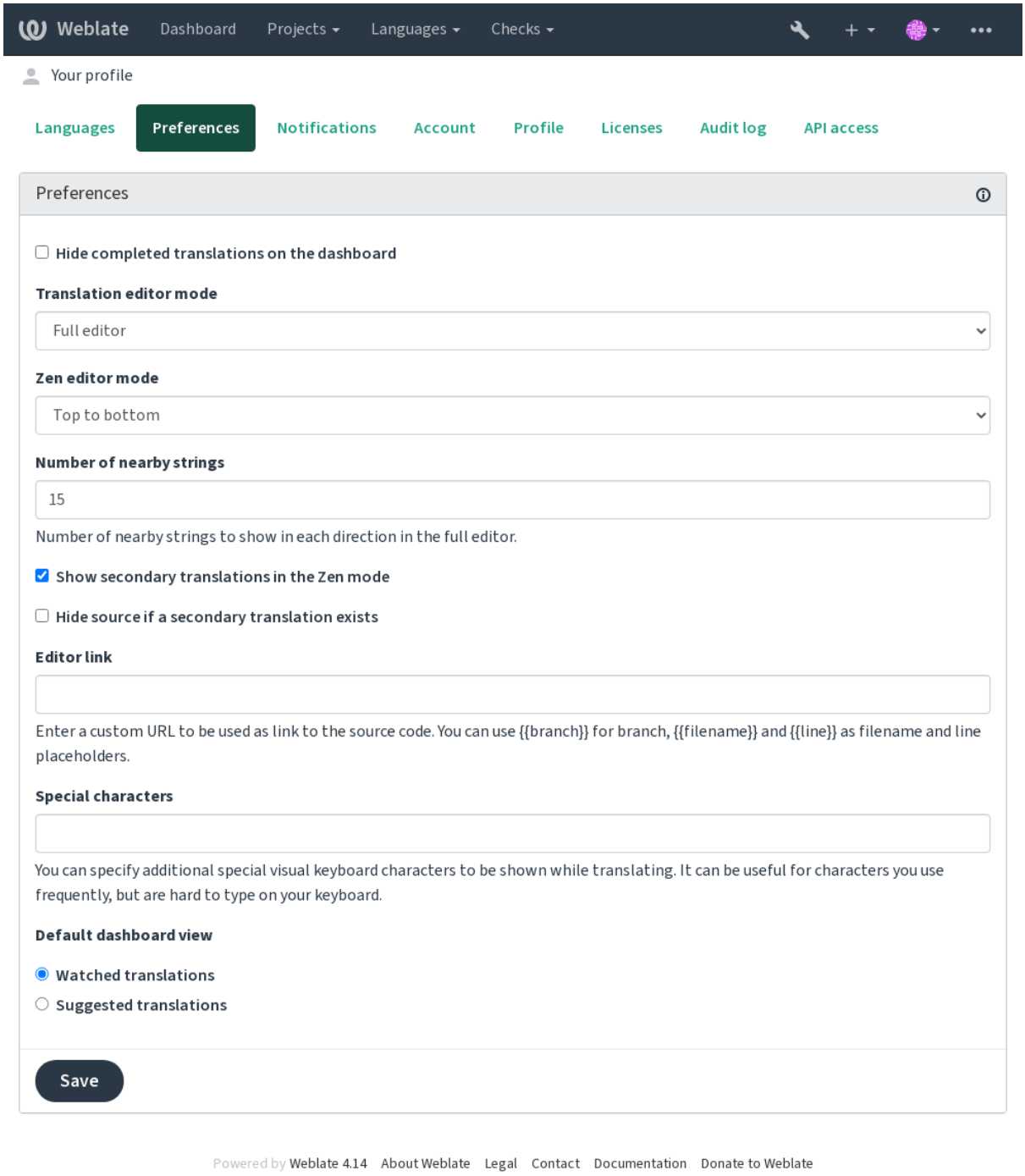
1.2.2 Painel

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

Novo na versão 2.5.

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

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



O menu tem estas opções:

- *Projetos* > *Visualizar todos os projetos* no menu principal mostra o estado da tradução, para cada projeto, na instância do Weblate.
- Selecionar um idioma no menu principal de *Idiomas* irá mostrar o estado da tradução de todos os projetos, filtrado por um dos seus idiomas primários.
- *Traduções observadas* no Painel vai mostrar o estado da tradução apenas dos projetos que está observando, filtradas por os seus idiomas primários.

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

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

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

1.2.3 Perfil do utilizador

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

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

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

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

Idiomas

1.2.4 Idioma da interface

Escolha o idioma que deseja exibir na UI.

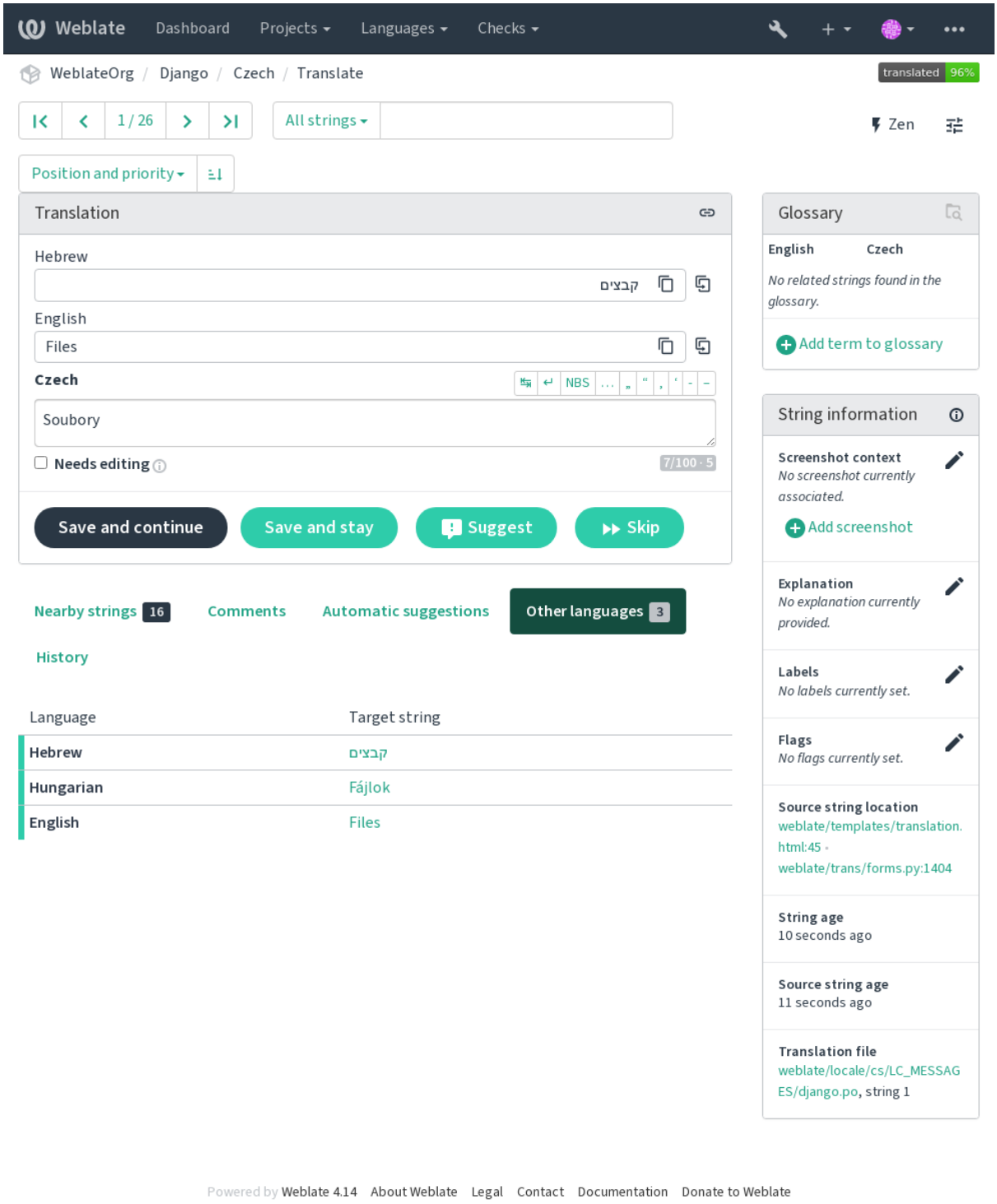
Idiomas traduzidos

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

Weblate
Dashboard
Projects
Languages
Checks
Dashboard
Watched translations 13
Suggested translations 5
Insights
Search
Translation
Translated
Unfinished
Unfinished words
Checks
Suggestions
WeblateOrg/Android — Czech
MIT
76%
3
3
WeblateOrg/Django — Hebrew
GPL-3.0
92%
2
15
WeblateOrg/Django — Czech
GPL-3.0
96%
1
12
4
WeblateOrg/Django — Hungarian
GPL-3.0
69%
8
109
1
WeblateOrg/Djangojs — Hebrew
GPL-3.0
WeblateOrg/Djangojs — Hungarian
GPL-3.0
96%
2
6
WeblateOrg/Djangojs — Czech
GPL-3.0
WeblateOrg/Language names — Czech
GPL-3.0
WeblateOrg/Language names — Hebrew
GPL-3.0
WeblateOrg/Language names — Hungarian
GPL-3.0
81%
4
5
WeblateOrg/WeblateOrg — Hebrew
GPL-3.0
WeblateOrg/WeblateOrg — Czech
GPL-3.0
WeblateOrg/WeblateOrg — Hungarian
GPL-3.0
Powered by Weblate 4.14
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Idiomas secundários

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



1.2.5 Preferências

Visualização predefinida do painel

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

Veja também:

Lista de componentes

Hiperligação do editor

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

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

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

Veja também:

Pode encontrar mais informações sobre o registo de protocolos de URL personalizados para o editor na documentação do Nette.

Carateres especiais

Additional special characters to include in the *Teclado visual*.

1.2.6 Notificações

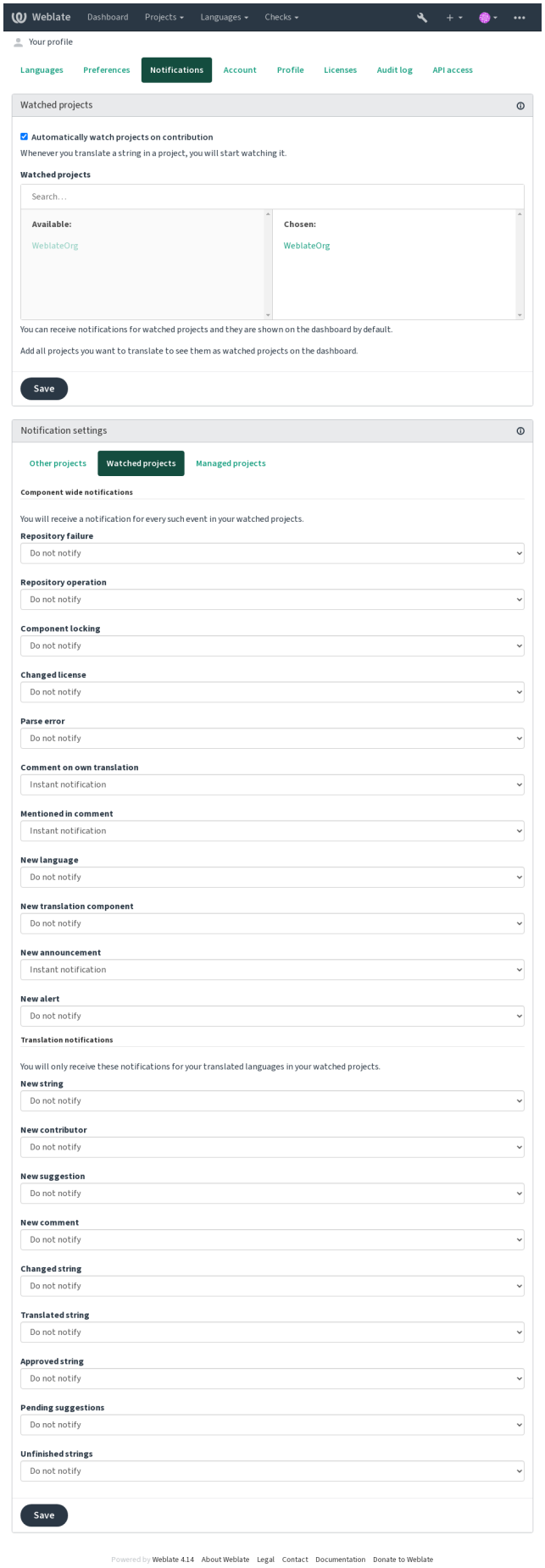
Inscreva-se em várias notificações da guia *Notificações*. As notificações para eventos selecionados em projetos assistidos ou administrados serão lhe enviadas por e-mail.

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

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

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

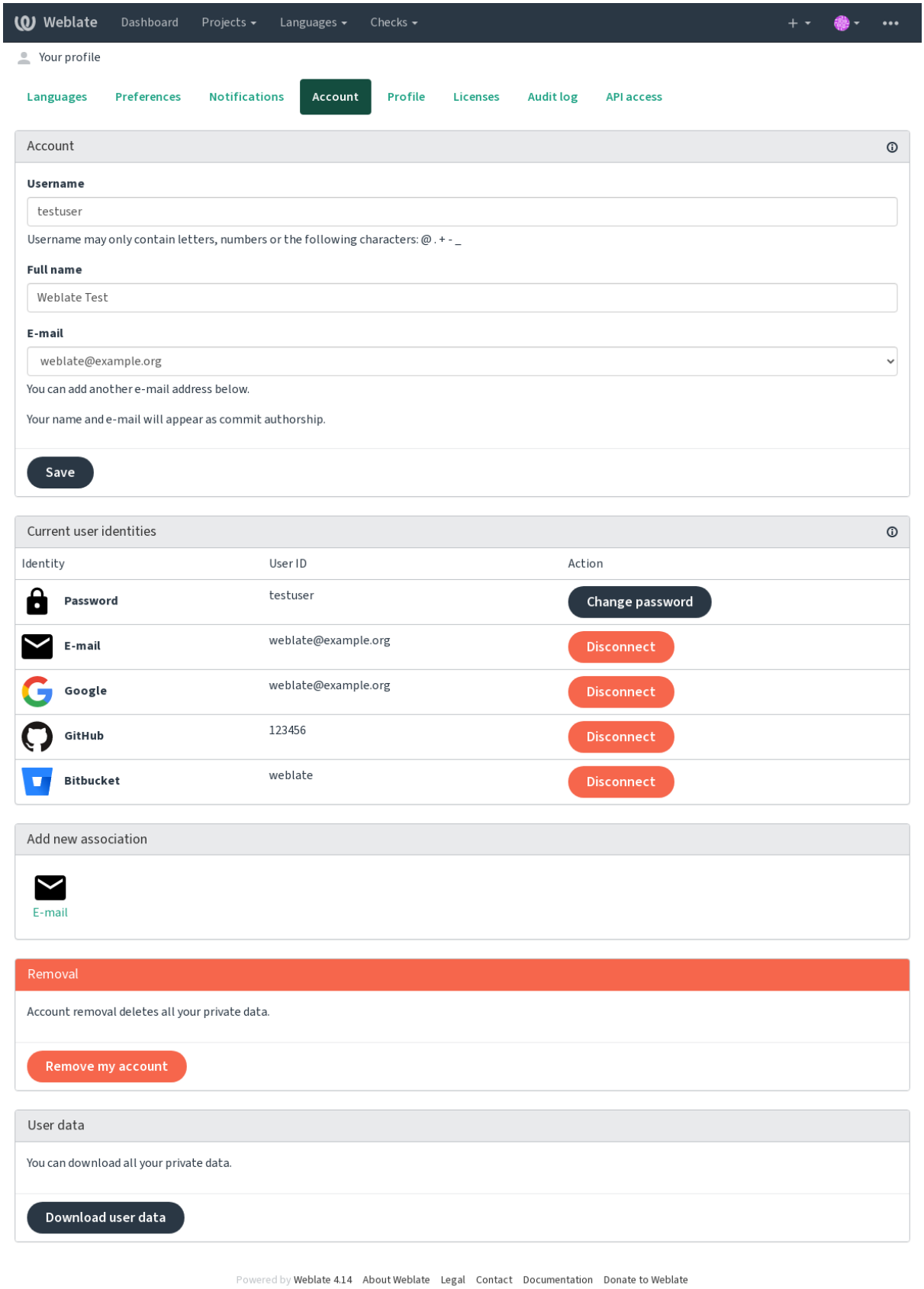
Nota: Não receberá notificações para as suas próprias ações.


Weblate
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
Unfinished strings
Do not notify
Save
Powered by Weblate 4.14
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.2.7 Conta

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

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


Weblate
Dashboard
Projects
Languages
Checks
Your profile
Languages
Preferences
Notifications
Account
Profile
Licenses
Audit log
API access
Account
Username
testuser
Username may only contain letters, numbers or the following characters: @ . + - _
Full name
Weblate Test
E-mail
weblate@example.org
You can add another e-mail address below.
Your name and e-mail will appear as commit authorship.
Save
Current user identities
Identity
User ID
Action
Password
testuser
Change password
E-mail
weblate@example.org
Disconnect
Google
weblate@example.org
Disconnect
GitHub
123456
Disconnect
Bitbucket
weblate
Disconnect
Add new association
E-mail
Removal
Account removal deletes all your private data.
Remove my account
User data
You can download all your private data.
Download user data
Powered by Weblate 4.14
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.2.8 Perfil

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

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

1.2.9 Licenças

1.2.10 Acesso API

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

1.2.11 Registo de auditoria

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

Veja também:

Executar por trás de um proxy reverso

1.3 Traduzir a usar o Weblate

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

No geral, há dois modos de tradução:

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

Por favor, veja *Fluxos de trabalho de tradução* para obter mais informações sobre fluxo de trabalho de tradução.

Opções para a visibilidade do projeto de tradução:

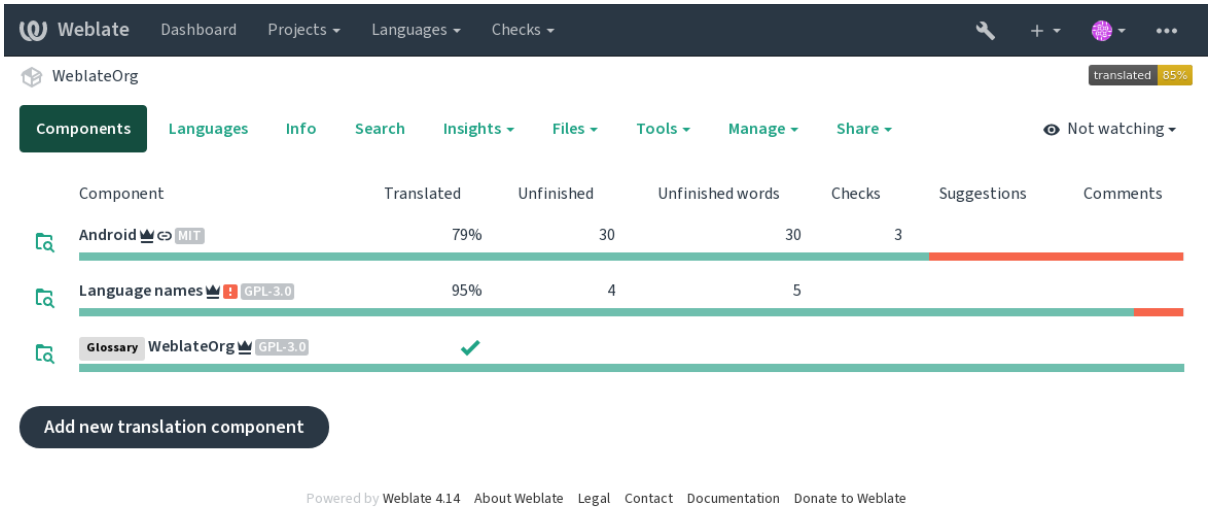
- Publicly visible
- Visível apenas para um certo grupo de tradutores

Veja também:

Controlo de acesso, *Fluxos de trabalho de tradução*

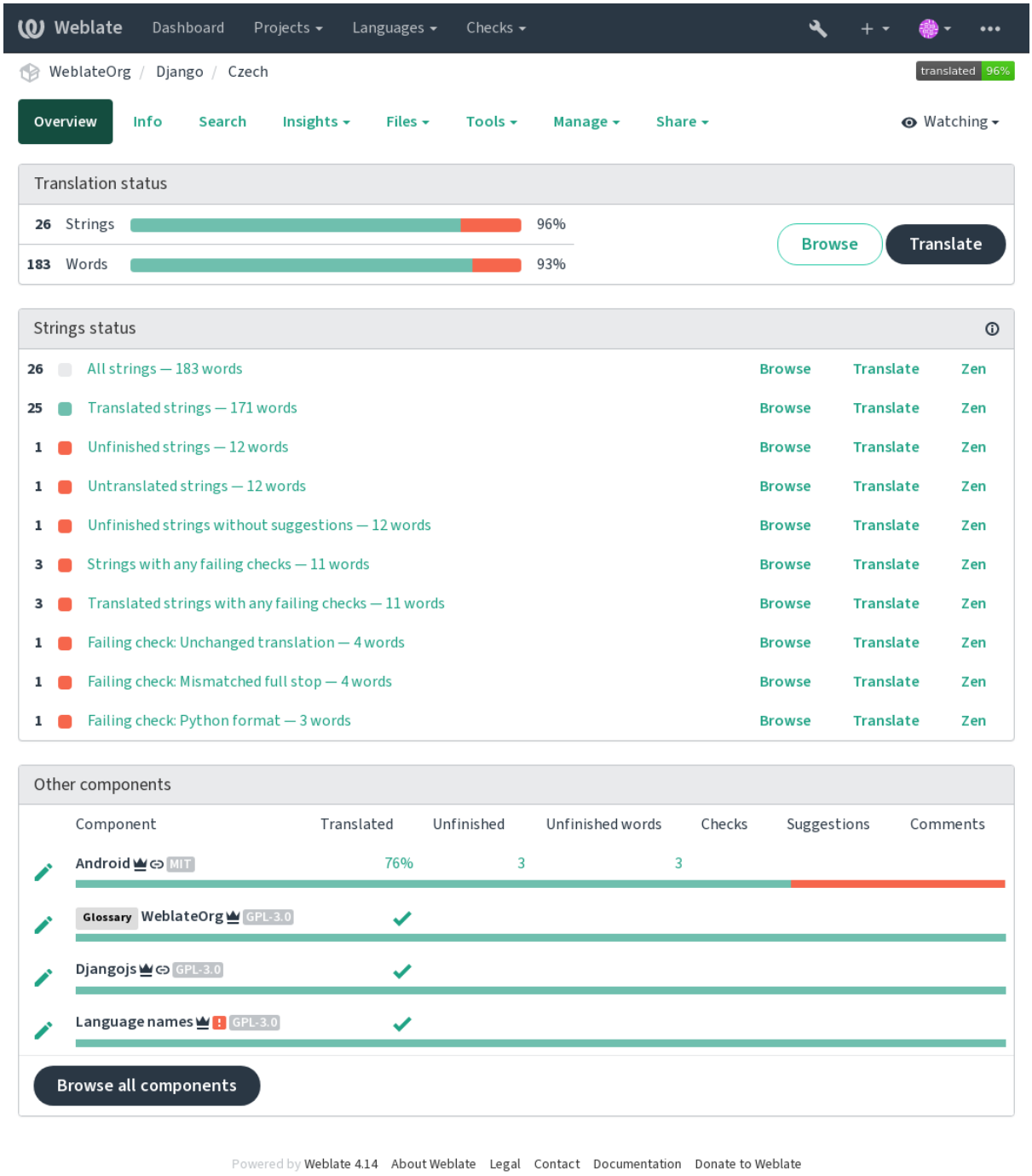
1.3.1 Projetos de tradução

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



1.3.2 Ligações de tradução

Having navigated to a component, a set of links lead to its actual translation. The translation is further divided into individual checks, like *Untranslated strings* or *Unfinished strings*. If the whole project is translated, without error, *All strings* is still available. Alternatively you can use the search field to find a specific string or term.



1.3.3 Sugestões

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

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

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

1.3.4 Comentários

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

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

Veja também:

report-source, *Revisões de cadeias fonte*, *Ativar revisões de fontes*

1.3.5 Variantes

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

Veja também:

variants, *Variantes*

1.3.6 Etiquetas

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

As seguintes etiquetas são usadas pelo Weblate:

Traduzido automaticamente
O texto foi traduzido usando *Tradução automática*.

Fonte precisa de revisão
A cadeia foi marcada para revisão utilizando :ref:`source-reviews".

Veja também:

labels

1.3.7 Traduzir

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

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

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

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

Plurais

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

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

Veja também:

Fórmula de plural

Alternative translations

Novo na versão 4.13.

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

With some formats, it is possible to have more translations for a single string. You can add more alternative translations using the *Tools* menu. Any blank alternative translations will be automatically removed upon saving.

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

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

Position and priority

Translation

English

Singular
%(count)s word

Plural
%(count)s words

Czech, One NBS ... „ " , ' - –
%(count)s slovo
15/140 · 14

Czech, Few NBS ... „ " , ' - –
%(count)s slova
15/140 · 15

Czech, Other NBS ... „ " , ' - –
%(count)s slov
14/140 · 15

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

Needs editing

Save and continue | Save and stay | Suggest | Skip

Nearby strings 20 | Comments | Automatic suggestions | Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Glossary

English Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
No explanation currently provided.

Labels
No labels currently set.

Flags
python-format

Source string location
weblate/templates/translation.html:149

String age
6 seconds ago

Source string age
6 seconds ago

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

Powered by Weblate 4.14 About Weblate Legal Contact Documentation Donate to Weblate

Atalhos de teclado

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

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

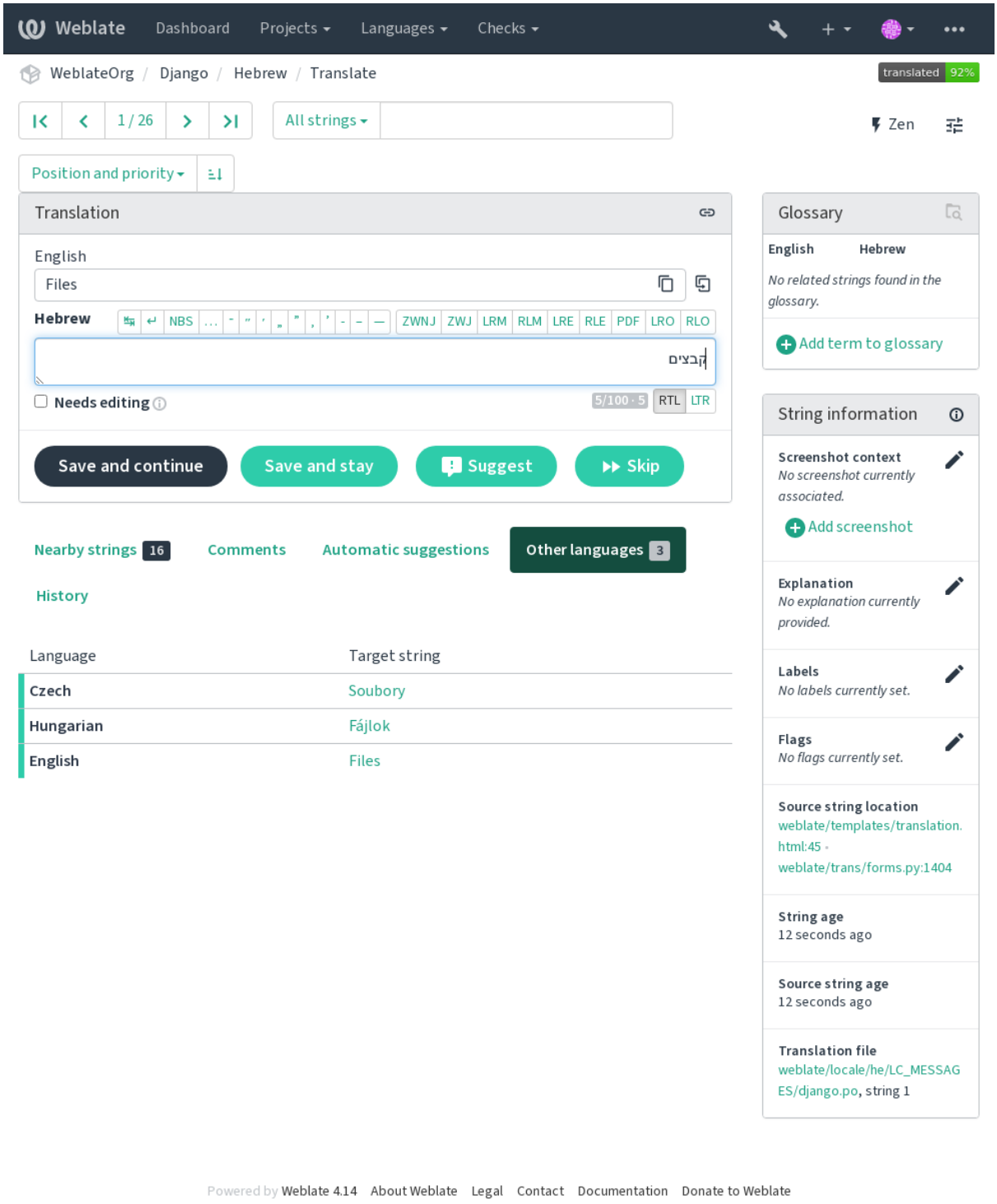
Atalho de teclado	Descrição
`Alt+Home`	De momento, esta tradução está bloqueada.
`Alt+End`	De momento, esta tradução está bloqueada.
`Alt+PageUp` ou `Ctrl+↑` or `Alt+↑` or `Cmd+↑`	De momento, esta tradução está bloqueada.
`Alt+PageDown` ou `Ctrl+↓` ou `Alt+↓` ou `Cmd+↓`	De momento, esta tradução está bloqueada.
`Alt+Enter` ou `Ctrl+Enter` ou `Cmd+Enter`	Envia o formulário atual; isso é o mesmo que pressionar *Gravar e continuar* enquanto edita a tradução.
`Ctrl+Shift+Enter` ou `Cmd+Shift+Enter`	Desmarca a tradução como a precisar de edição e a envia.
`Ctrl+E` ou `Cmd+E`	Muda o foco ao editor de tradução.
`Ctrl+U` ou `Cmd+U`	Muda o foco ao editor de comentários.
`Ctrl+M` ou `Cmd+M`	Mostra a guia *Sugestões automáticas*, consulte *Sugestões automáticas*.
`Ctrl+1` até `Ctrl+9` ou `Cmd+1` até `Cmd+9`	Copia objetos colocáveis de determinada quantidade da cadeia fonte.
`Ctrl+M+1` até `9` ou `Cmd+M+1` até `9`	É passado como um parâmetro único que consiste o nome de uma tradução atual.
`Ctrl+I+1` até `9` ou `Cmd+I+1` até `9`	Ignore um item na lista de verificações falhadas.
`Ctrl+J` ou `Cmd+J`	Mostra a guia de *Cadeias próximas*.
`Ctrl+S` ou `Cmd+S`	Muda o foco para o campo de pesquisa.
`Ctrl+O` ou `Cmd+O`	Copia a cadeia fonte.
`Ctrl+Y` ou `Cmd+Y`	Toggle the *Needs editing* checkbox.

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:

- User configured *Carateres especiais* defined in the *Perfil do utilizador*
- Caracteres por idioma fornecidos pelo Weblate (por exemplo, citações ou caracteres específicos RTL)
- Caracteres configurados a usar `SPECIAL_CHARS`


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Hebrew / Translate
translated 92%
1 / 26
All strings
Zen
Position and priority
Translation
English
Files
Hebrew
NBS ... ZWNJ ZWJ LRM RLM LRE RLE PDF LRO RLO
קבצים
Needs editing
5/100 · 5
RTL LTR
Save and continue
Save and stay
Suggest
Skip
Nearby strings 16
Comments
Automatic suggestions
Other languages 3
History
Language
Target string
Czech
Soubory
Hungarian
Fájlok
English
Files
Glossary
English
Hebrew
No related strings found in the glossary.
Add term to glossary
String information
Screenshot context
No screenshot currently associated.
Add screenshot
Explanation
No explanation currently provided.
Labels
No labels currently set.
Flags
No flags currently set.
Source string location
weblate/templates/translation.html:45 · weblate/trans/forms.py:1404
String age
12 seconds ago
Source string age
12 seconds ago
Translation file
weblate/locale/he/LC_MESSAGES/django.po, string 1
Powered by Weblate 4.14
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Contexto da tradução

Esta descrição contextual fornece informações relacionadas sobre a cadeia atual.

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

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

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

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

Memória de tradução
Veja cadeias semelhantes traduzidas no passado, veja *Memória de Tradução*.

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

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

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

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

Histórico de tradução

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

Comprimento da cadeia traduzida

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

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

1.3.8 Sugestões automáticas

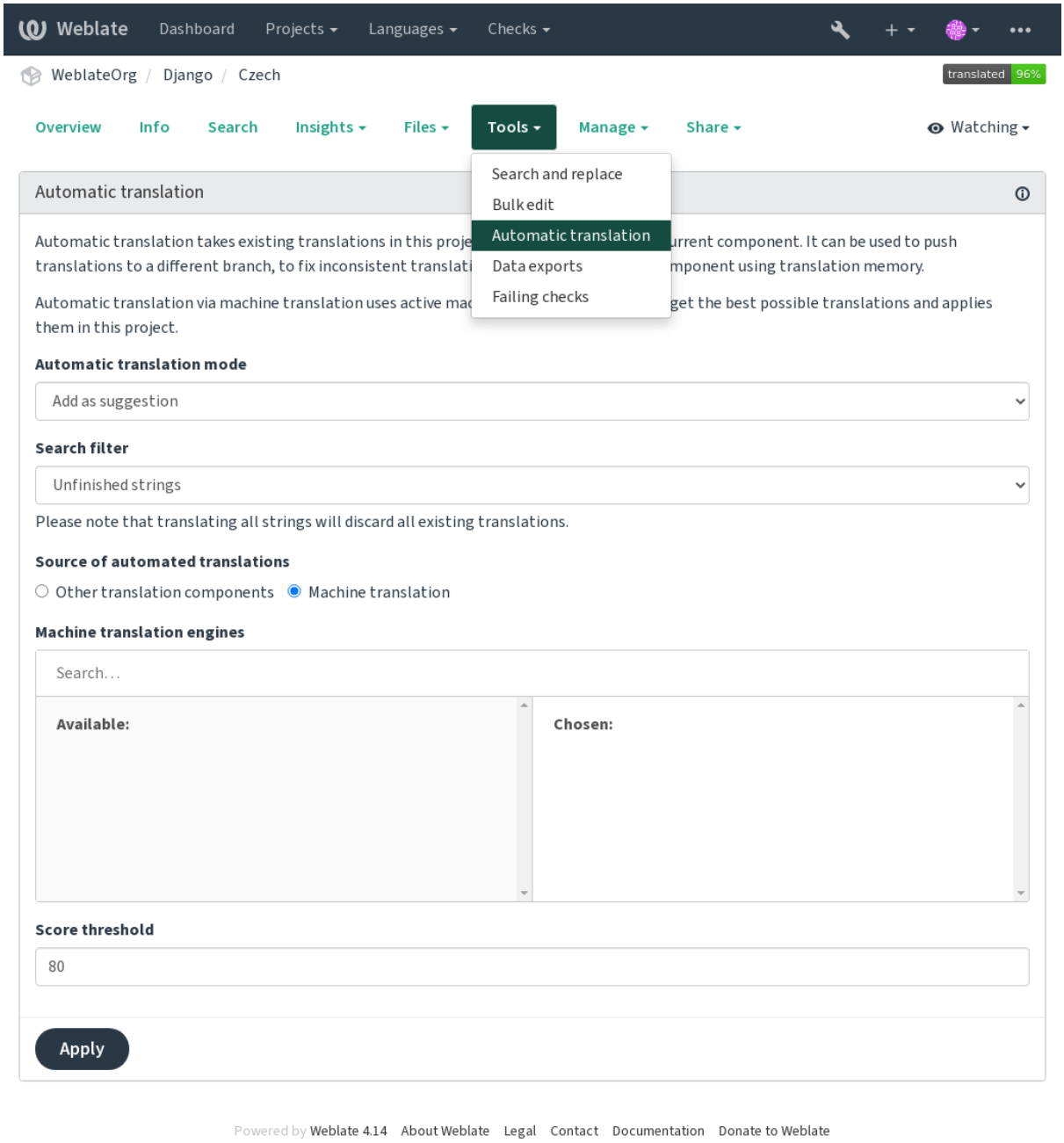
Baseado na configuração e o seu idioma traduzido, o Weblate fornece sugestões de várias ferramentas de tradução e memoria-tradução. Todas as traduções de máquina estão disponíveis numa única guia de cada página de tradução.

Veja também:

Encontra a lista de ferramentas suportadas em *Configuring automatic suggestions*.

1.3.9 Tradução automática

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



Dois modos de operação são possíveis:

- Usar outros componentes do Weblate como fonte para traduções.

- Usar serviços selecionados de tradução automática com traduções acima de um certo limite de qualidade.

Também pode escolher quais cadeias devem ser traduzidas automaticamente.

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

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

The automatically translated strings are labelled *Automatically translated*.

Veja também:

Manter traduções iguais entre componentes

1.3.10 Limitação de taxa

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

Os limites predefinidos e o ajuste fino estão descritos no manual administrativo, veja *Limitação de taxa*.

1.3.11 Procurar e substituir

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

Dica: Não te preocupes em mexer nas cadeias. Este é um processo de dois passos que mostra uma pré-visualização das cadeias editadas antes que a alteração real seja confirmada.

1.3.12 Edição em massa

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

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

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

Veja também:

Extensão de edição em massa

1.3.13 Visualização em matriz

Para comparar diferentes idiomas de forma eficiente, pode usar a visualização em matriz. Ele está disponível em todas as páginas do componente no menu *Ferramentas*. Primeiro selecione todos os idiomas que deseja comparar e confirme a sua seleção, depois disso, pode clicar em qualquer tradução para abri-la e editá-la rapidamente.

A visualização em matriz também é um bom ponto de partida para encontrar traduções ausentes em diferentes idiomas e adicioná-las rapidamente a partir de uma visualização.

1.3.14 Modo zen

O editor Zen pode ser ativado a clicar no botão *Zen* no canto superior direito durante a tradução de um componente. Ele simplifica o layout e remove elementos adicionais da IU, como *Textos próximos* ou *Glossário*.

Pode selecionar o editor Zen como seu editor padrão a usar a guia *Preferências* no seu: *Perfil do utilizador*. Aqui também pode escolher entre ter traduções listadas *De cima para baixo* ou *Lado a lado* a depender da sua preferência pessoal.

1.4 Descarregar e enviar traduções

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

Nota: As opções disponíveis podem ser limitadas por configurações de *controle de acesso*.

1.4.1 Descarregar traduções

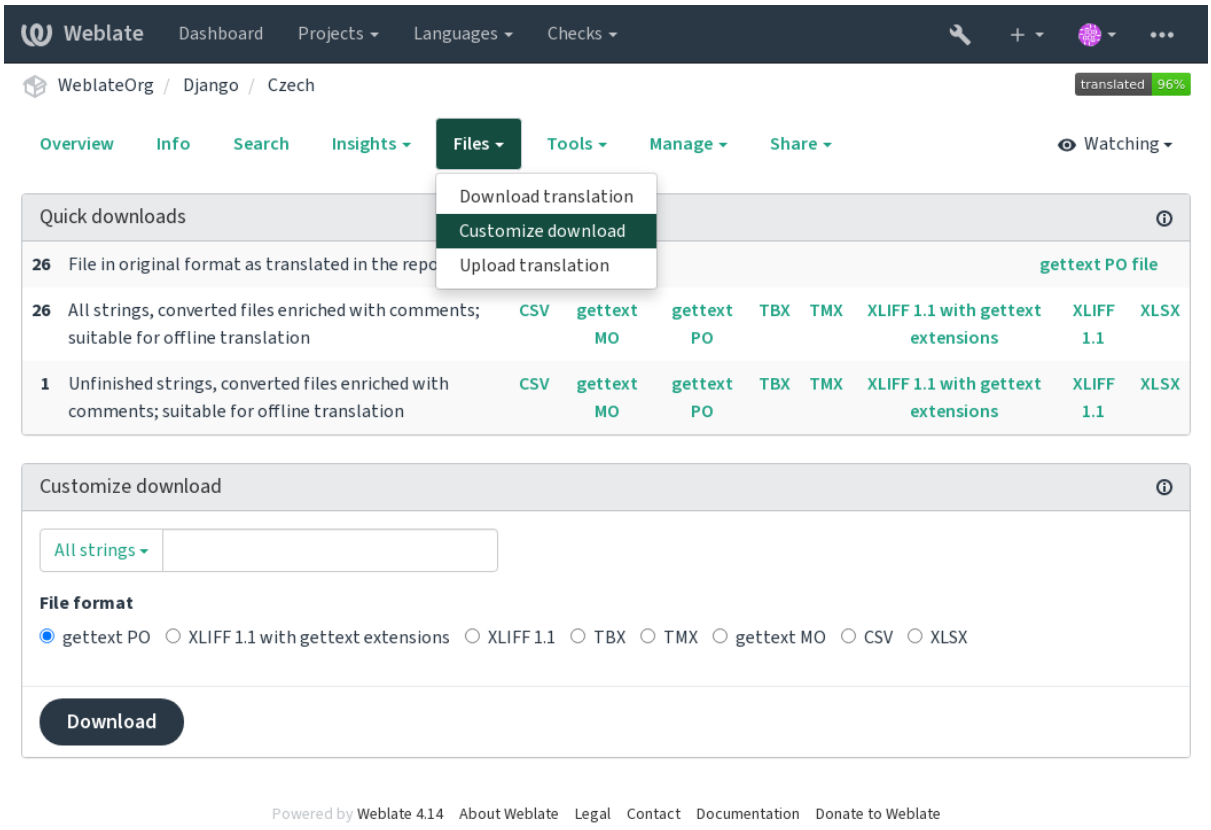
A partir do painel do projeto ou componente, ficheiros traduzíveis podem ser descarregados no menu *Ficheiros*.

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

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

- gettext PO
- XLIFF com extensões gettext
- XLIFF 1.1
- TermBase eXchange
- Translation Memory eXchange
- gettext MO (disponível apenas quando a tradução está usando gettext PO)
- CSV
- Excel Open XML
- JSON (disponível apenas para traduções monolíngues)
- Recurso de textos de Android (disponível apenas para traduções monolíngues)
- Textos do iOS (disponível apenas para traduções monolíngues)

Dica: O conteúdo disponível nos ficheiros convertidos difere com base nos recursos de formato de arquivo, pode encontrar uma visão geral em *Capacidades dos tipos de tradução*.

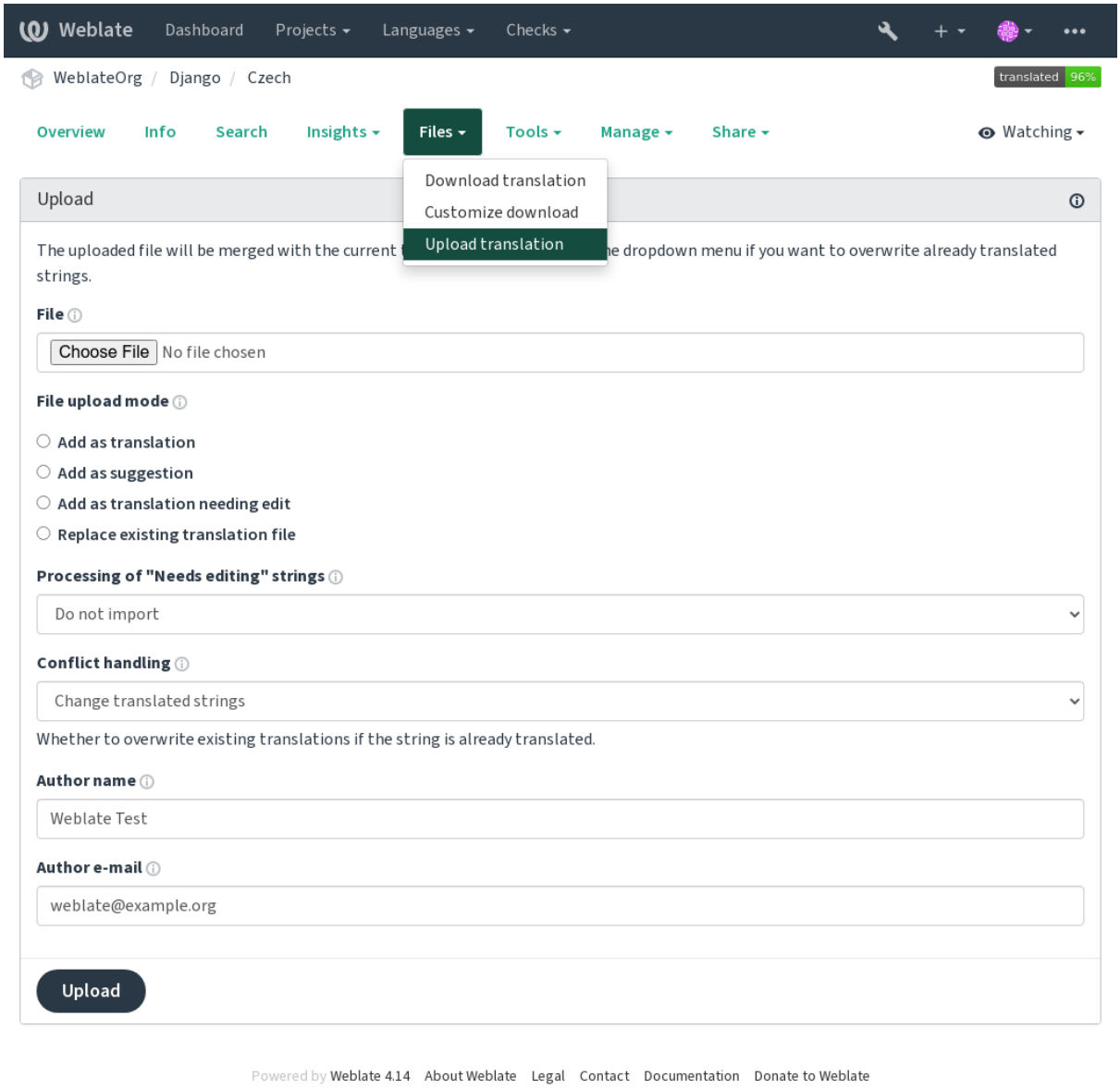


Veja também:

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

1.4.2 Enviar traduções

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



Formatos de ficheiros suportados

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

Veja também:

Formatos de ficheiros suportados, *Descarregar e enviar traduções*

Métodos de importação

Estas são as opções apresentadas ao enviar ficheiros de tradução:

Adicionar como tradução (`translate`)
Os textos importados são adicionados como traduções para textos existentes. Este é o caso de uso mais comum e o comportamento padrão.

Apenas traduções são usadas do ficheiro carregado e nenhum conteúdo adicional.

Adicionar como sugestão (`suggest`)
Os textos importados são adicionados como sugestões, faça isso quando quiser ter os seus textos enviados revisados.

Apenas traduções são usadas do ficheiro carregado e nenhum conteúdo adicional.

Adicionar como tradução que necessita de edição («fuzzy»)
Os textos importados são adicionados como traduções que necessitam de edição. Isso pode ser útil quando quer que as traduções sejam usadas, mas também revisadas.

Apenas traduções são usadas do ficheiro carregado e nenhum conteúdo adicional.

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

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

Esta opção é suportada apenas para alguns formatos de ficheiro.

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

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

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

Apenas fonte, tradução e chave (contexto) são usados do ficheiro enviado.

Veja também:

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

Gestão de conflitos

Define como lidar com cadeias enviadas que já são traduzidas.

Cadeias necessitando de edição

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

Substituindo autoria

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

1.5 Glossário

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

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

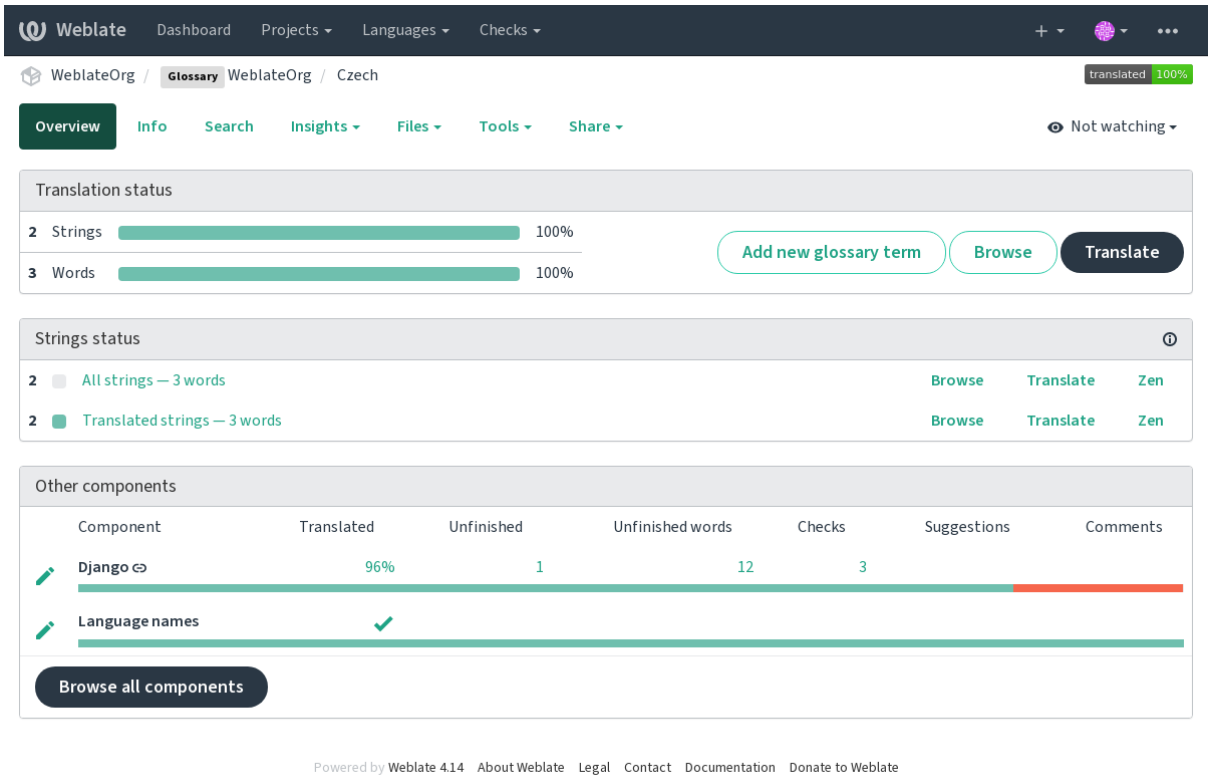
1.5.1 Gestão de glossários

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

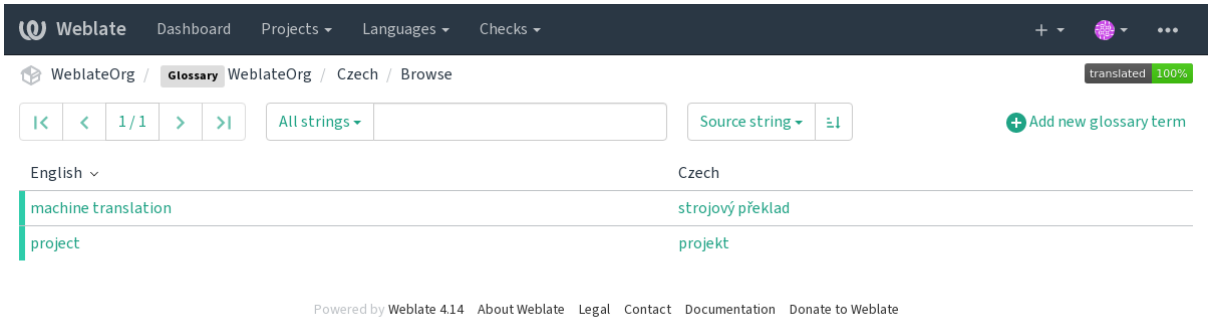
Use qualquer componente como um glossário a ligar *Usar como um glossário*. Pode criar vários glossários para um projeto.

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

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



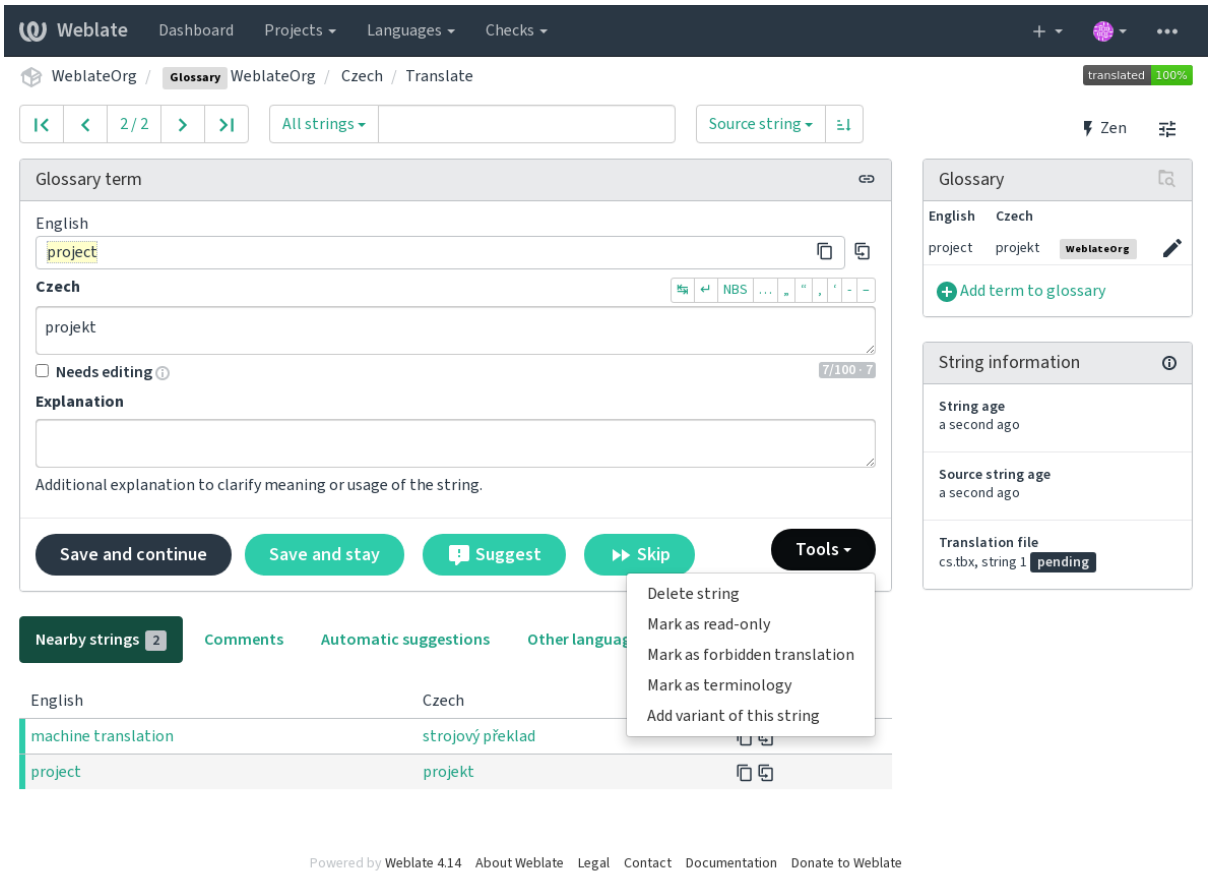
Pode navegar por todos os termos do glossário:



ou edite-os como qualquer tradução.

1.5.2 Termos de glossário

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



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 untranslatable* means they can not be translated. Use this for brand names or other terms that should not be changed in other languages. Such terms are visually highlighted in the glossary sidebar.

Veja também:

Personalizando comportamento a usar marcadores

Traduções proibidas

Novo na versão 4.5.

Marcar certas traduções de termos do glossário como `forbidden`, editando em massa, digitando o marcador, ou usando *Ferramentas ↓ Marcar como tradução proibida* significa que **não** devem ser usado. Use isto para esclarecer a tradução quando algumas palavras são ambíguas ou podem ter significados inesperados.

Veja também:

Personalizando comportamento a usar marcadores

Terminologia

Novo na versão 4.5.

Marcar certos termos do glossário como `terminology` editando em massa, digitando o marcador ou usando *Ferramentas ↓ Marcar como terminologia* adiciona entradas para eles em todos os idiomas no glossário. Use-o para termos importantes que devem ser bem pensados e manter um significado consistente em todos os idiomas.

Veja também:

Personalizando comportamento a usar marcadores

Variantes

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

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

Veja também:

variants

1.6 Verificações e correções

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

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

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

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

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

Translation

English

Singular
%(count)s word

Plural
%(count)s words

Czech, One

0/140 · 14

Czech, Few
několik slov

12/140 · 15

Czech, Other
%(count)s slov

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

14/140 · 15

Needs editing

Save and continue Save and stay Suggest Skip

Nearby strings 20 Comments Automatic suggestions Other languages 3 History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format 1
Following format strings are missing: %(count)s
Dismiss
For all languages

Missing plurals 2
Some plural forms are untranslated
Dismiss
For all languages

Glossary

English Czech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
No explanation currently provided.

Labels
No labels currently set.

Flags
python-format

Source string location
weblate/templates/translation.html:149

String age
9 seconds ago

Source string age
10 seconds ago

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

Powered by Weblate 4.14 About Weblate Legal Contact Documentation Donate to Weblate

1.6.1 Correções automáticas

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

Veja também:

`AUTOFIX_LIST`

1.6.2 Verificações de qualidade

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

Veja também:

`CHECK_LIST`, *Personalizando comportamento a usar marcadores*

1.6.3 Verificações de tradução

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

Markup BBCode

Resumo
O BBCode na tradução não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.BBCodeCheck`

Check identifier
`bbcode`

Marcador para ignorar
`ignore-bbcode`

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

Esta verificação garante que eles também estejam na tradução.

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

Palavras consecutivas duplicadas

Novo na versão 4.1.

Resumo
O texto contém a mesma palavra duas vezes seguidas:

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.duplicate.DuplicateCheck`

Check identifier
`duplicate`

Marcador para ignorar
`ignore-duplicate`

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

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

Não segue o glossário

Novo na versão 4.5.

Resumo
A tradução não segue os termos definidos num glossário.

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.glossary.GlossaryCheck`

Check identifier
`check_glossary`

Sinalize para ativar
`check-glossary`

Marcador para ignorar
`ignore-check-glossary`

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

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

Veja também:

Glossário, *Personalizando comportamento a usar marcadores*, *Marcadores de tradução*

Espaço duplo

Resumo
A tradução contém espaço duplo

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.DoubleSpaceCheck`

Check identifier
`double_space`

Marcador para ignorar
`ignore-double-space`

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

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

Cadeias formatadas

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

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

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

Dica: Caso a verificação de formato específico não esteja disponível no Weblate, pode usar *Espaços reservados* genéricos.

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

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

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

Position and priority

Translation

English

Singular
%(count)s word

Plural
%(count)s words

Czech, One
%(count)s slovo
15/140 · 14

Czech, Few
%(count)s slova
15/140 · 15

Czech, Other
%(count)s slov
14/140 · 15

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

Needs editing

Save and continue Save and stay Suggest Skip

Nearby strings 20 Comments Automatic suggestions Other languages 3

History

None
String updated in the repository
WeblateOrg / Django — Czech

English
Singular
%(count)s word
Plural
%(count)s words
Czech 48 characters edited Current translation Translated
One
%(count)s slovo
Few
%(count)s slova
Other
%(count)s slov
6 seconds ago

Glossary
English Czech
No related strings found in the glossary.
Add term to glossary

String information
Screenshot context
No screenshot currently associated.
Add screenshot
Explanation
No explanation currently provided.
Labels
No labels currently set.
Flags
python-format
Source string location
weblate/templates/translation.html:149
String age
6 seconds ago
Source string age
6 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5

Browse all component changes

Powered by Weblate 4.14 About Weblate Legal Contact Documentation Donate to Weblate

Cadeia de interpolação AngularJS

Resumo
As cadeias de interpolação AngularJS não correspondem à fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.angularjs.AngularJSInterpolationCheck`

Check identifier
`angularjs_format`

Sinalize para ativar
`angularjs-format`

Marcador para ignorar
`ignore-angularjs-format`

Exemplo de texto de formato nomeado
`O seu saldo é {{amount}} {{ currency }}`

Veja também:

Cadeias formatadas, Interpolação de texto do AngularJS

Formato C

Resumo
O formato C da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.CFormatCheck`

Check identifier
`c_format`

Sinalize para ativar
`c-format`

Marcador para ignorar
`ignore-c-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Exemplo de texto de formato de posição
`O seu saldo é %1$d %2$s`

Veja também:

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

Formato C#

Resumo
O formato C# da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.CSharpFormatCheck`

Check identifier
`c_sharp_format`

Sinalize para ativar
`c-sharp-format`

Marcador para ignorar
`ignore-c-sharp-format`

Exemplo de texto de formato de posição
`Há {0} maçãs`

Veja também:

Cadeias formatadas, Formato de string do C#

Literais de modelo de ECMAScript

Resumo
As literais de modelo ECMAScript não correspondem com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.ESTemplateLiteralsCheck`

Check identifier
`es_format`

Sinalize para ativar
`es-format`

Marcador para ignorar
`ignore-es-format`

Exemplo de interpolação
`Há ${number} maçãs`

Veja também:

Cadeias formatadas, Template strings

Interpolação de i18next

Novo na versão 4.0.

Resumo
A interpolação de i18next não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.I18NextInterpolationCheck`

Check identifier
`i18next_interpolation`

Sinalize para ativar
`i18next-interpolation`

Marcador para ignorar
`ignore-i18next-interpolation`

Exemplo de interpolação
`Há {{number}} maçãs`

Exemplo de aninhamento
`Há $t(number) maçãs`

Veja também:

Cadeias formatadas, Interpolação i18next

MessageFormat do ICU

Novo na versão 4.9.

Resumo
Erros de sintaxe e/ou incompatibilidades de espaços reservados em textos MessageFormat do ICU.

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.icu.ICUMessageFormatCheck`

Check identifier
`icu_message_format`

Sinalize para ativar
`icu-message-format`

Marcador para ignorar
`ignore-icu-message-format`

Exemplo de interpolação
`Há {number, plural, one {uma maçã} other {# maçãs}}.`

Essa verificação tem suporte para mensagens ICU MessageFormat puras, bem como ICU com tags XML simples. Pode configurar o comportamento desta verificação a usar `icu-flags:*`, ou a optar pelo suporte XML ou a desativar certas sub-verificações. Por exemplo, a seguinte sinalização ativa o suporte a XML enquanto desativa a validação de submensagens plurais:

`xml`	Ativa o suporte para tags XML simples. Por padrão, as tags XML são analisadas livremente. Caracteres < perdidos são ignorados se não forem razoavelmente parte de uma tag.
`strict-xml`	Ativa o suporte para tags XML estritas. Todos os caracteres < devem ser escapados se não fizerem parte de uma tag.
`-highlight`	Desativa o realce de espaços reservados no editor.
`-require_other`	Desativa a exigência de submensagens ter um seletor `other`.
`-submessage_selector`	Ignora a verificação de que os seletores de submensagem correspondem à fonte.
`-types`	Ignora a verificação de que os tipos de espaço reservado correspondem à fonte.
`-extra`	Ignora a verificação de nenhum espaço reservado está presente que não esteva presente no texto fonte.
`-missing`	Ignora a verificação de que não falta nenhum espaço reservado que estava presente no texto fonte.

Além disso, quando `strict-xml` não está ativado, mas `xml` está ativado, pode usar o marcador `icu-tag-prefix: PREFIXO` para exigir que todas as tags XML comecem com um texto específico. Por exemplo, o seguinte marcador só permitirá que as tags XML sejam correspondidas se começarem com `<x:`:

Isso corresponderia a `<x:link>clique aqui</x:link>`, mas não a `<strong>isso</strong>`.

Veja também:

Sintaxe de MessageFormat do ICU, *Cadeias formatadas*, ICU: Formatação de mensagens, Format.JS: sintaxe de mensagens

Formato Java

Resumo
O formato Java da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.JavaFormatCheck`

Check identifier
`java_printf_format`

Sinalize para ativar
`java-printf-format`

Marcador para ignorar
`ignore-java-printf-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Exemplo de texto de formato de posição
`O seu saldo é %1$d %2$s`

Alterado na versão 4.14: This used to be toggled by `java-format` flag, it was changed for consistency with GNU gettext.

Veja também:

Cadeias formatadas, Java Format Strings

Formato de Mensagem Java

Resumo
O MessageFormat Java da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.JavaMessageFormatCheck`

Check identifier
`java_format`

Marcador para ativar incondicionalmente
`java-format`

Marcador para ativar autodetecção
`auto-java-messageformat` ativa verificação somente se houver um texto de formato na fonte

Marcador para ignorar
`ignore-java-format`

Exemplo de texto de formato de posição
`Há {0} maçãs`

Alterado na versão 4.14: This used to be toggled by `java-messageformat` flag, it was changed for consistency with GNU gettext.

Veja também:

Cadeias formatadas, Java MessageFormat

Formato JavaScript

Resumo
O formato JavaScript da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.JavaScriptFormatCheck`

Check identifier
`javascript_format`

Sinalize para ativar
`javascript-format`

Marcador para ignorar
`ignore-javascript-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Veja também:

Cadeias formatadas, JavaScript formatting strings

Formato Lua

Resumo
O formato de cadeia Lua não corresponde com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.LuaFormatCheck`

Check identifier
`lua_format`

Sinalize para ativar
`lua-format`

Marcador para ignorar
`ignore-lua-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Veja também:

Cadeias formatadas, Lua formatting strings

Formato Object Pascal

Resumo
O formato Object Pascal da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.ObjectPascalFormatCheck`

Check identifier
`object_pascal_format`

Sinalize para ativar
`object-pascal-format`

Marcador para ignorar
`ignore-object-pascal-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Veja também:

Cadeias formatadas, Object Pascal formatting strings, Free Pascal formatting strings Delphi formatting strings

Espaços reservados de percentagem

Novo na versão 4.0.

Resumo
Os símbolos de percentagem não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.PercentPlaceholdersCheck`

Check identifier
`percent_placeholders`

Sinalize para ativar
`percent-placeholders`

Marcador para ignorar
`ignore-percent-placeholders`

Exemplo de texto de formato simples
`Há %number% maçãs`

Veja também:

Cadeias formatadas,

Formato Perl

Resumo
O formato Perl da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.PerlFormatCheck`

Check identifier
`perl_format`

Sinalize para ativar
`perl-format`

Marcador para ignorar
`ignore-perl-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Exemplo de texto de formato de posição
`O seu saldo é %1$d %2$s`

Veja também:

Cadeias formatadas, Perl sprintf, Perl Format Strings

Formato PHP

Resumo
O formato PHP da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.PHPFormatCheck`

Check identifier
`php_format`

Sinalize para ativar
`php-format`

Marcador para ignorar
`ignore-php-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Exemplo de texto de formato de posição
`O seu saldo é %1$d %2$s`

Veja também:

Cadeias formatadas, Documentação de PHP sprintf, PHP Format Strings

Formato de chaveta Python

Resumo
O formato de chaveta Python da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.PythonBraceFormatCheck`

Check identifier
`python_brace_format`

Sinalize para ativar
`python-brace-format`

Marcador para ignorar
`ignore-python-brace-format`

Cadeia de formato simples
`Há {} maçãs`

Exemplo de texto de formato nomeado
`O seu saldo é {amount} {currency}`

Veja também:

Cadeias formatadas, Formato de chaves Python, Python Format Strings

Formato Python

Resumo
O formato Python da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.PythonFormatCheck`

Check identifier
`python_format`

Sinalize para ativar
`python-format`

Marcador para ignorar
`ignore-python-format`

Cadeia de formato simples
`Há %d maçãs`

Exemplo de texto de formato nomeado
`O seu saldo é %(amount)d %(currency)s`

Veja também:

Cadeias formatadas, Formatação de texto Python, Python Format Strings

Formato Qt

Resumo
O formato Qt da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.qt.QtFormatCheck`

Check identifier
`qt_format`

Sinalize para ativar
`qt-format`

Marcador para ignorar
`ignore-qt-format`

Exemplo de texto de formato de posição
`Há %1 maçãs`

Veja também:

Cadeias formatadas, Qt QString::arg()

Forma plural Qt

Resumo
O formato plural Qt da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.qt.QtPluralCheck`

Check identifier
`qt_plural_format`

Sinalize para ativar
`qt-plural-format`

Marcador para ignorar
`ignore-qt-plural-format`

Exemplo de texto de formato de plural
`Há %Ln maçã(s)`

Veja também:

Cadeias formatadas, Guia de i18n do Qt

Formato Ruby

Resumo
O formato Ruby da cadeia não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.ruby.RubyFormatCheck`

Check identifier
`ruby_format`

Sinalize para ativar
`ruby-format`

Marcador para ignorar
`ignore-ruby-format`

Exemplo de texto de formato simples
`Há %d maçãs`

Exemplo de texto de formato de posição
`O seu saldo é %1$f %2$s`

Exemplo de texto de formato nomeado
`O seu saldo é %+.2<amount>f %<currency>s`

Cadeia de modelo nomeado
`O seu saldo é %{amount} %{currency}`

Veja também:

Cadeias formatadas, Ruby Kernel#sprintf

Formato Scheme

Resumo
O formato de cadeia Scheme não corresponde com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.SchemeFormatCheck`

Check identifier
`scheme_format`

Sinalize para ativar
`scheme-format`

Marcador para ignorar
`ignore-scheme-format`

Exemplo de texto de formato simples
`Há ~d maçãs`

Veja também:

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

Formatação vue I18n

Resumo
A formatação vue I18n não corresponde à fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.format.VueFormattingCheck`

Check identifier
`vue_format`

Sinalize para ativar
`vue-format`

Marcador para ignorar
`ignore-vue-format`

Formatação nomeada
`Há {count} maçãs`

Formatação i18n de Rails
`Há %{count} maçãs`

Mensagens de localidade vinculadas
`@:message.dio @:message.the_world!`

Veja também:

Cadeias formatadas, Formatação Vue I18n, Mensagens de localidade Vue I18n

Foi traduzido

Resumo
Esta cadeia foi traduzida anteriormente

Âmbito
todos os textos

Classe da verificação
`weblate.checks.consistency.TranslatedCheck`

Check identifier
`translated`

Marcador para ignorar
`ignore-translated`

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

Inconsistente

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

Âmbito
todos os textos

Classe da verificação
`weblate.checks.consistency.ConsistencyCheck`

Check identifier
`inconsistent`

Marcador para ignorar
`ignore-inconsistent`

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

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

Esta verificação aplica-se a todos os componentes de um projeto que tenham *Permitir propagação da tradução* ativados.

Dica: Por motivos de desempenho, a verificação pode não encontrar todas as inconsistências, ela limita a quantidade de correspondências.

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

Pode usar *Tradução automática* para automatizar a tradução de textos recém-adicionados que já estão traduzidos em outro componente.

Veja também:

Manter traduções iguais entre componentes

Letra Kashida utilizada

Novo na versão 3.5.

Resumo
As letras decorativas kashida não deverão ser utilizadas

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.KashidaCheck`

Check identifier
`kashida`

Marcador para ignorar
`ignore-kashida`

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

Veja também:

Kashida na Wikipédia

Hiperligações de marcação

Novo na versão 3.5.

Resumo
As hiperligações de marcação não coincidem com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.MarkdownLinkCheck`

Check identifier
`md-link`

Sinalize para ativar
`md-text`

Marcador para ignorar
`ignore-md-link`

As ligações de Markdown não correspondem com a fonte.

Veja também:

Ligações Markdown

Referências de Markdown

Novo na versão 3.5.

Resumo
As referências de hiperligação de marcação não correspondem à fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.MarkdownRefLinkCheck`

Check identifier
`md-reflink`

Sinalize para ativar
`md-text`

Marcador para ignorar
`ignore-md-reflink`

As referências de ligação Markdown não correspondem à fonte.

Veja também:

Ligações Markdown

Sintaxe de Markdown

Novo na versão 3.5.

Resumo
A sintaxe de Markdown não coincide com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.MarkdownSyntaxCheck`

Check identifier
`md-syntax`

Sinalize para ativar
`md-text`

Marcador para ignorar
`ignore-md-syntax`

A sintaxe de Markdown não coincide com a fonte

Veja também:

Elementos span Markdown

Tamanho máximo da tradução

Resumo
A tradução não deve exceder o tamanho dado

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.MaxLengthCheck`

Check identifier
`max-length`

Sinalize para ativar
`max-length`

Marcador para ignorar
`ignore-max-length`

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

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

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

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

Quando o marcador `xml-text` também é usado, o cálculo do comprimento ignora as tags XML.

Tamanho máximo da tradução

Resumo
A tradução não deve exceder o tamanho dado

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.render.MaxSizeCheck`

Check identifier
`max-size`

Sinalize para ativar
`max-size`

Marcador para ignorar
`ignore-max-size`

Novo na versão 3.7.

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

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

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

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

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

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

Quando o marcador `xml-text` também é usado, o cálculo do comprimento ignora as tags XML.

Veja também:

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

\n não correspondente

Resumo
Number of \n literals in translation does not match source

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EscapedNewlineCountingCheck`

Check identifier
`escaped_newline`

Marcador para ignorar
`ignore-escaped-newline`

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

Dois pontos não correspondentes

Resumo
A fonte e a tradução não terminam ambas com uma vírgula

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndColonCheck`

Check identifier
`end_colon`

Marcador para ignorar
`ignore-end-colon`

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

Veja também:

Dois pontos na Wikipédia

Reticências não correspondentes

Resumo
A fonte e a tradução não terminam ambas com reticências

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndEllipsisCheck`

Check identifier
`end_ellipsis`

Marcador para ignorar
`ignore-end-ellipsis`

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

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

Veja também:

Reticências na Wikipédia

Ponto de exclamação não correspondente

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

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndExclamationCheck`

Check identifier
`end_exclamation`

Marcador para ignorar
`ignore-end-exclamation`

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

Veja também:

Ponto de exclamação na Wikipédia

Ponto final não correspondente

Resumo
A fonte e a tradução não terminam ambas com ponto

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndStopCheck`

Check identifier
`end_stop`

Marcador para ignorar
`ignore-end-stop`

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

Veja também:

Ponto final na Wikipédia

Ponto de interrogação não correspondente

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

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndQuestionCheck`

Check identifier
`end_question`

Marcador para ignorar
`ignore-end-question`

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

Veja também:

Ponto de interrogação na Wikipédia

Ponto e vírgula não correspondente

Resumo
A fonte e a tradução não terminam ambas com ponto e vírgula

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndSemicolonCheck`

Check identifier
`end_semicolon`

Marcador para ignorar
`ignore-end-semicolon`

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

Veja também:

Ponto e vírgula na Wikipédia

Quebras de linha não coincidentes

Resumo
A quantidade de novas linhas na tradução não corresponde ao da fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.NewLineCountCheck`

Check identifier
`newline-count`

Marcador para ignorar
`ignore-newline-count`

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

Faltam plurais

Resumo
Alguns plurais não estão traduzidos

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.consistency.PluralsCheck`

Check identifier
`plurals`

Marcador para ignorar
`ignore-plurals`

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

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

Espaços reservados

Novo na versão 3.9.

Resumo
Faltam alguns espaços reservados na tradução

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.placeholders.PlaceholderCheck`

Check identifier
`placeholders`

Sinalize para ativar
`placeholders`

Marcador para ignorar
`ignore-placeholders`

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

Alterado na versão 4.13: With the `case-insensitive` flag, the placeholders are not case-sensitive.

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

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

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

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

You can also have case insensitive placeholders:

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

Veja também:

Personalizando comportamento a usar marcadores

Espaçamento da pontuação

Novo na versão 3.9.

Resumo
Falta de espaço não quebrável antes do sinal de pontuação duplo

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.PunctuationSpacingCheck`

Check identifier
`punctuation_spacing`

Marcador para ignorar
`ignore-punctuation-spacing`

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

Veja também:

Espaçamento em francês e inglês na Wikipédia

Expressão regular

Novo na versão 3.9.

Resumo
A tradução não corresponde à expressão regular

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.placeholders.RegexCheck`

Check identifier
`regex`

Sinalize para ativar
`regex`

Marcador para ignorar
`ignore-regex`

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

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

Mesmos plurais

Resumo
Alguns plurais estão traduzidos da mesma maneira

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.consistency.SamePluralsCheck`

Check identifier
`same-plurals`

Marcador para ignorar
`ignore-same-plurals`

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

Resumo
A fonte e a tradução não iniciam ambas com uma nova linha

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.BeginNewlineCheck`

Check identifier
`begin_newline`

Marcador para ignorar
`ignore-begin-newline`

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

Veja também:

Nova linha no final

Espaços no início

Resumo
A fonte e a tradução não iniciam ambas com a mesma quantidade de espaços

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.BeginSpaceCheck`

Check identifier
`begin_space`

Marcador para ignorar
`ignore-begin-space`

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

Nova linha no final

Resumo
A fonte e a tradução não terminam ambas com uma nova linha

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndNewlineCheck`

Check identifier
`end_newline`

Marcador para ignorar
`ignore-end-newline`

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

Veja também:

Nova linha no início

Espaço no final

Resumo
A fonte e a tradução não terminam ambas com um espaço

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.EndSpaceCheck`

Check identifier
`end_space`

Marcador para ignorar
`ignore-end-space`

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

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

Tradução inalterada

Resumo
Fonte e tradução são iguais

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.same.SameCheck`

Check identifier
`same`

Marcador para ignorar
`ignore-same`

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

Esta verificação pode ajudar a encontrar cadeias erroneamente não traduzidos.

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

Veja também:

Configuração de componente, *Personalizando comportamento a usar marcadores*

HTML inseguro

Novo na versão 3.9.

Resumo
A tradução usa marcação HTML insegura

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.SafeHTMLCheck`

Check identifier
`safe-html`

Sinalize para ativar
`safe-html`

Marcador para ignorar
`ignore-safe-html`

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

Dica: Quando o marcador `md-text` também é usado, as ligações de estilo Markdown também são permitidas.

Veja também:

A verificação HTML é feita pela biblioteca Bleach desenvolvida pela Mozilla.

URL

Novo na versão 3.5.

Resumo
A tradução não contém um URL

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.URLCheck`

Check identifier
`url`

Sinalize para ativar
`url`

Marcador para ignorar
`ignore-url`

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

Markup XML

Resumo
Etiquetas XML na tradução não coincidem com a fonte

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.XMLTagsCheck`

Check identifier
`xml-tags`

Marcador para ignorar
`ignore-xml-tags`

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.

Nota: Esta verificação é desativada pelo marcador `safe-html`, pois a limpeza de HTML feita por ele pode produzir marcação HTML que não é um XML válido.

Sintaxe XML

Novo na versão 2.8.

Resumo
A tradução não é XML válido

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.markup.XMLValidityCheck`

Check identifier
`xml-invalid`

Marcador para ignorar
`ignore-xml-invalid`

A marcação XML não é válida.

Nota: Esta verificação é desativada pelo marcador `safe-html`, pois a limpeza de HTML feita por ele pode produzir marcação HTML que não é um XML válido.

Espaçamento nulo

Resumo
A tradução contém carateres extra de espaço nulo

Âmbito
textos traduzidos

Classe da verificação
`weblate.checks.chars.ZeroWidthSpaceCheck`

Check identifier
`zero-width-space`

Marcador para ignorar
`ignore-zero-width-space`

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

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

Veja também:

Espaço de largura zero na Wikipédia

1.6.4 Verificações de texto fonte

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

Reticências

Resumo
A cadeia utiliza três pontos (…) em vez do carácter de reticências (…)

Âmbito
entradas fonte

Classe da verificação
`weblate.checks.source.EllipsisCheck`

Check identifier
`ellipsis`

Marcador para ignorar
`ignore-ellipsis`

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

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

Veja também:

Reticências na Wikipédia

Sintaxe de MessageFormat do ICU

Novo na versão 4.9.

Resumo
Erros de sintaxe em textos de MessageFormat do ICU.

Âmbito
entradas fonte

Classe da verificação
`weblate.checks.icu.ICUSourceCheck`

Check identifier
`icu_message_format_syntax`

Sinalize para ativar
`icu-message-format`

Marcador para ignorar
`ignore-icu-message-format`

Veja também:

MessageFormat do ICU

Não traduzido há muito tempo

Novo na versão 4.1.

Resumo
A cadeia não foi traduzida há muito tempo

Âmbito
entradas fonte

Classe da verificação
`weblate.checks.source.LongUntranslatedCheck`

Check identifier
`long_untranslated`

Marcador para ignorar
`ignore-long-untranslated`

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

Várias verificações falhadas

Resumo
As traduções em várias línguas têm verificações falhadas

Âmbito
entradas fonte

Classe da verificação
`weblate.checks.source.MultipleFailingCheck`

Check identifier
`multiple_failures`

Marcador para ignorar
`ignore-multiple-failures`

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

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

Várias variáveis sem nome

Novo na versão 4.1.

Resumo
Há várias variáveis sem nome na cadeia, o que faz impossível aos tradutores reordená-las

Âmbito
entradas fonte

Classe da verificação
`weblate.checks.format.MultipleUnnamedFormatsCheck`

Check identifier
`unnamed_format`

Marcador para ignorar
`ignore-unnamed-format`

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

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

Não pluralizado

Resumo
A cadeia é utilizada como plural, mas não tem plural

Âmbito
entradas fonte

Classe da verificação
`weblate.checks.source.OptionalPluralCheck`

Check identifier
`optional_plural`

Marcador para ignorar
`ignore-optional-plural`

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

Por exemplo, com o Gettext em Python poderia ser:

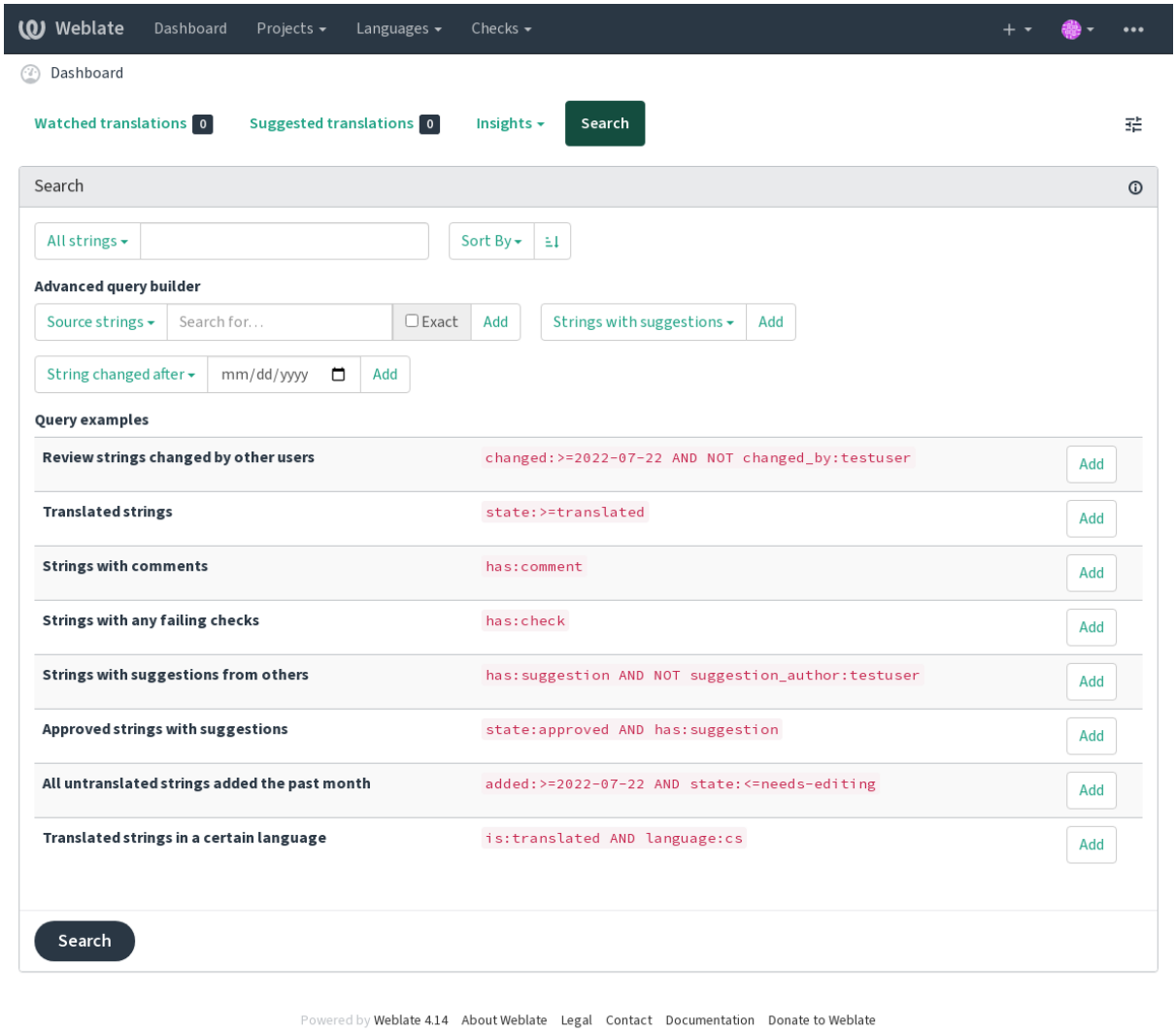
```
from gettext import ngettext

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

1.7 Pesquisando

Novo na versão 3.9.

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

When no field is defined, the lookup happens on source, target, and context strings.



1.7.1 Pesquisa simples

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

1.7.2 Campos

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

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

`priority:NÚMERO`
Prioridade da cadeia.

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

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

`pending:BOOLEANO`
Cadeia pendente para descarga para o VCS.

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

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

`language:TEXTO`
Idioma de destino da cadeia.

`component:TEXTO`
URL amigável de componente ou pesquisa que não diferencia maiúsculas de minúsculas, veja *URL amigável do componente* e *Nome do componente*.

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

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

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

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

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

`check:TEXTO`
String has failing check, see *Verificações e correções* for check identifiers.

`dismissed_check:TEXTO`
String has dismissed check, see *Verificações e correções* for check identifiers.

`comment:TEXTO`
Pesquisa nos comentários de utilizadores.

`resolved_comment:TEXT`
Search in resolved comments.

`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.

`label:TEXT`
Search in labels.

`screenshot:TEXT`
Search in screenshots.

1.7.3 Operadores booleanos

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

1.7.4 Operações de campo

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

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

`changed:2019`
Alterado no ano de 2019.

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

1.7.5 Operadores exatos

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

1.7.6 Pesquisando por alterações

Novo na versão 4.4.

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

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

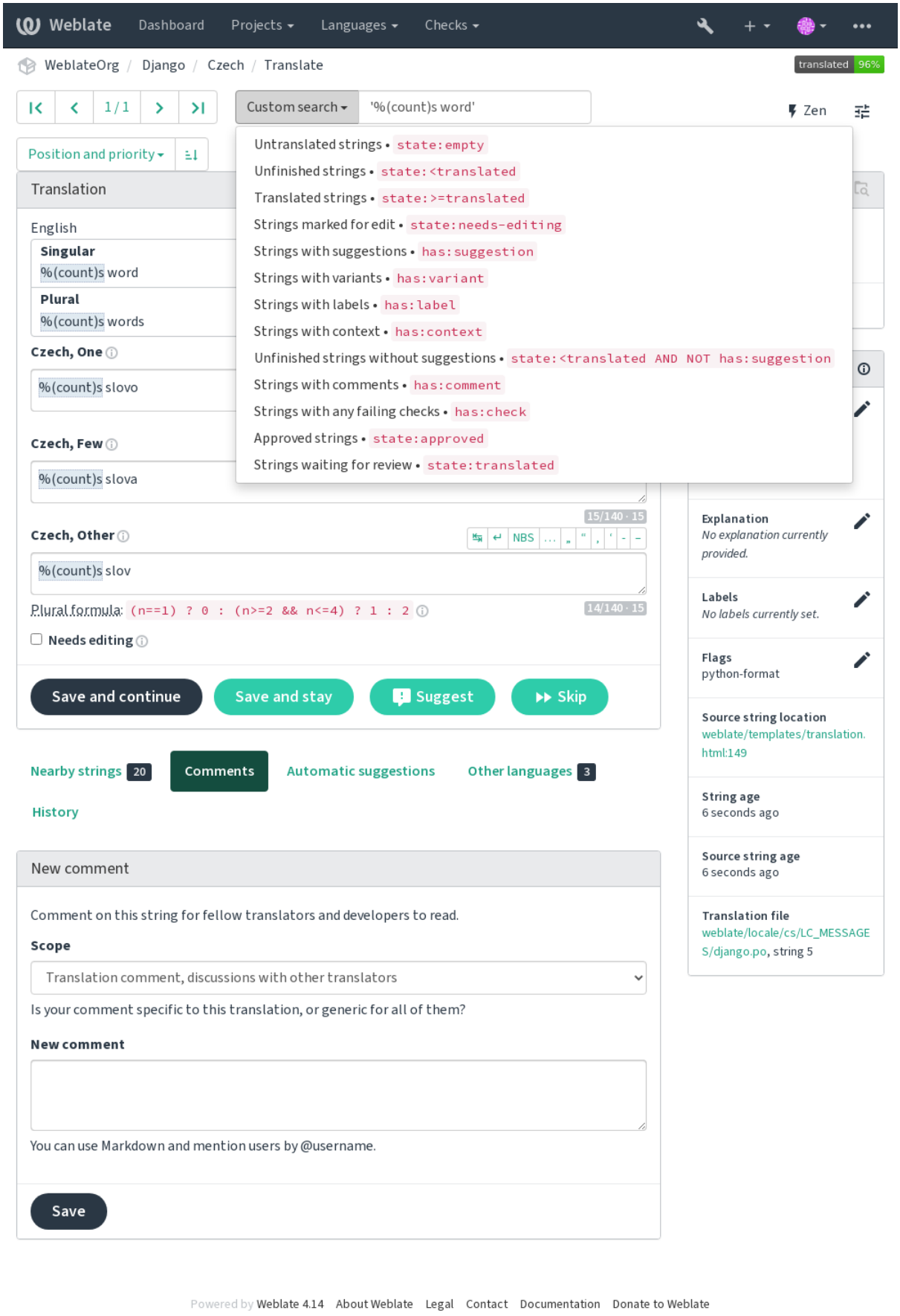
1.7.7 Expressões regulares

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

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

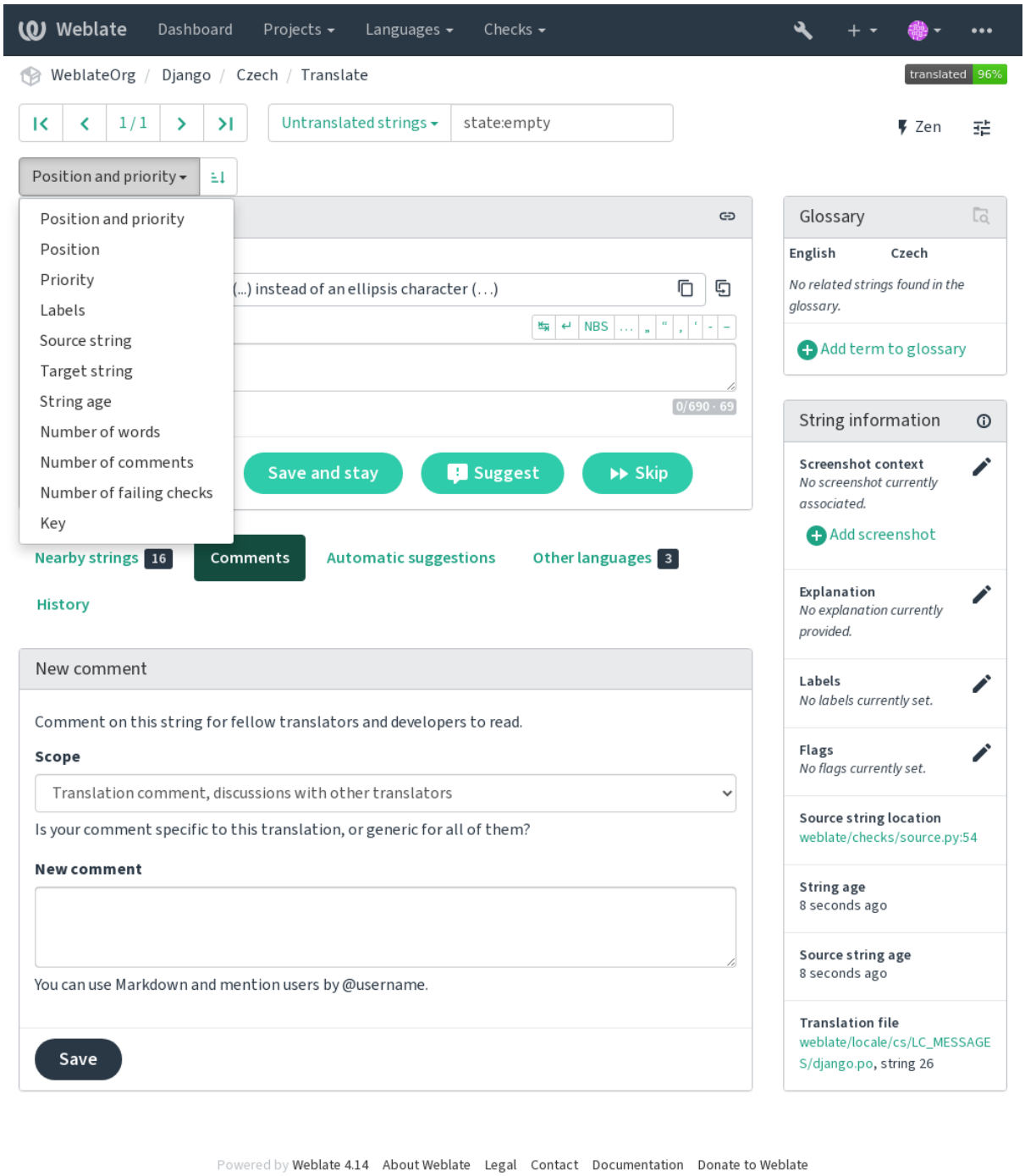
1.7.8 Consultas predefinidas

Pode selecionar 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
Position and priority
Untranslated strings • state:empty
Unfinished strings • state:<translated
Translated strings • state:>=translated
Strings marked for edit • state:needs-editing
Strings with suggestions • has:suggestion
Strings with variants • has:variant
Strings with labels • has:label
Strings with context • has:context
Unfinished strings without suggestions • state:<translated AND NOT has:suggestion
Strings with comments • has:comment
Strings with any failing checks • has:check
Approved strings • state:approved
Strings waiting for review • state:translated
Translation
English
Singular
%(count)s word
Plural
%(count)s words
Czech, One
%(count)s slovo
Czech, Few
%(count)s slova
15/140 · 15
Czech, Other
%(count)s slov
Plural formula: (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
14/140 · 15
Needs editing
Save and continue
Save and stay
Suggest
Skip
Nearby strings 20
Comments
Automatic suggestions
Other languages 3
History
New comment
Comment on this string for fellow translators and developers to read.
Scope
Translation comment, discussions with other translators
Is your comment specific to this translation, or generic for all of them?
New comment
You can use Markdown and mention users by @username.
Save
Explanation
No explanation currently provided.
Labels
No labels currently set.
Flags
python-format
Source string location
weblate/templates/translation.html:149
String age
6 seconds ago
Source string age
6 seconds ago
Translation file
weblate/locale/cs/LC_MESSAGES/django.po, string 5
Powered by Weblate 4.14
About Weblate
Legal
Contact
Documentation
Donate to Weblate

1.7.9 Ordenando os resultados

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



1.8 Fluxos de trabalho de tradução

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

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

1.8.1 Acesso à tradução

O *controle de acesso* não é discutido em detalhe como um todo nos fluxos de trabalho, pois a maioria das opções dele pode ser aplicada a qualquer fluxo de trabalho. Consulte a respectiva documentação sobre como gerir o acesso às traduções.

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

1.8.2 Estados de tradução

Cada cadeia traduzida pode estar num dos seguintes estados:

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

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

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

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

Este estado está disponível apenas quando revisões estão ativadas.

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

Os estados são representados nos ficheiros de tradução quando possível.

Dica: No caso do formato de ficheiro que utiliza não ter suporte ao armazenamento de estados, pode utilizar a extensão *Marcar as traduções inalteradas como «Precisa de edição»* para marcar textos inalterados como precisando de edição.

Veja também:

Capacidades dos tipos de tradução, *Fluxos de trabalho de tradução*

1.8.3 Tradução direta

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

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

Configuração	Valor	Nota
Activar revisões	inativo	Configurado a nível de projeto.
Ativar sugestões	ativo	É útil para os utilizadores serem capazes de sugerir quando não têm certeza.
Votação de sugestão	inativo	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	*Utilizadores*	Ou *Traduzir* com *controle de acesso por projeto*.
Grupo de revisores	N/D	Não usado.

1.8.4 Revisão por pares

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

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

Configuração	Valor	Nota
Activar revisões	inativo	Configurado a nível de projeto.
Ativar sugestões	ativo	
Votação de sugestão	inativo	
Aceitar sugestões automaticamente	1	Pode definir um valor mais alto para exibir mais revisões por pares.
Grupo de tradutores	*Utilizadores*	Ou *Traduzir* com *controle de acesso por projeto*.
Grupo de revisores	N/D	Não usado, todos os tradutores revisam.

1.8.5 Revisores dedicados

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

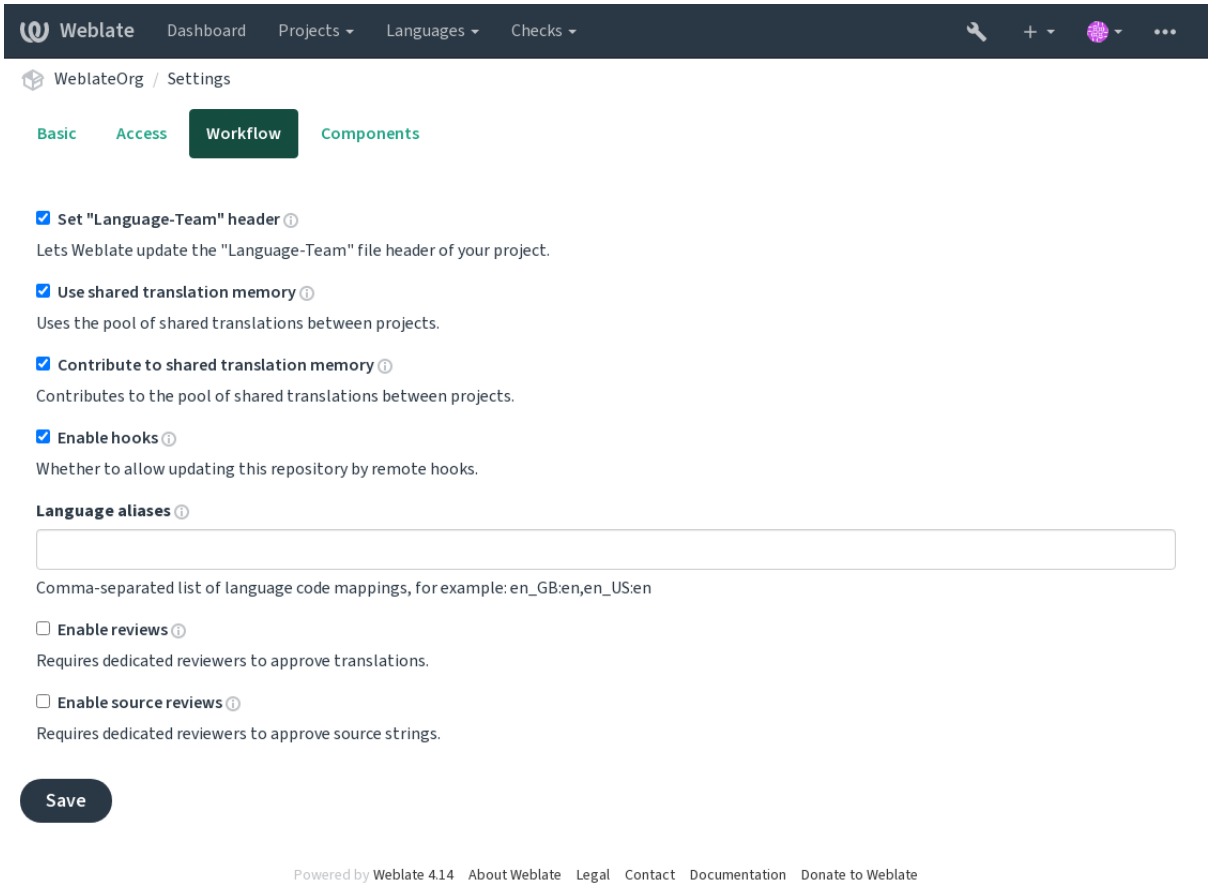
Com revisores dedicados tem dois grupos de utilizadores, um capaz de enviar traduções e outro capaz de revisá-los para garantir que as traduções sejam consistentes e que a qualidade seja boa.

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

Configuração	Valor	Nota
Activar revisões	ativo	Configurado a nível de projeto.
Ativar sugestões	inativo	É útil para os utilizadores serem capazes de sugerir quando não têm certeza.
Votação de sugestão	inativo	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	*Utilizadores*	Ou *Traduzir* com *controle de acesso por projeto*.
Grupo de revisores	*Revisores*	Ou *Revisar* com *controle de acesso por projeto*.

1.8.6 Ativar revisões

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

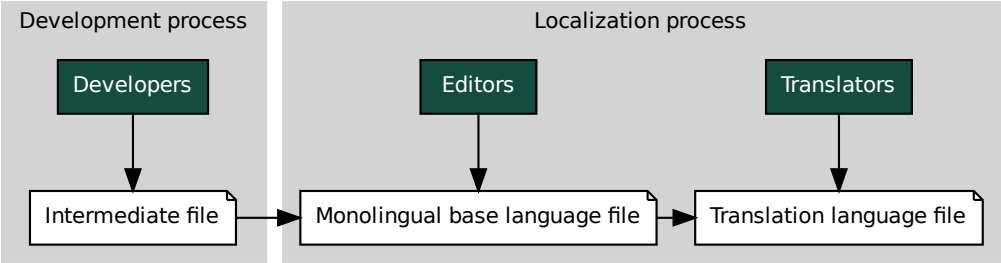


Nota: Dependendo da configuração do Weblate, a configuração pode não estar-lhe disponível. Por exemplo, no Hosted Weblate, isso não está disponível para projetos hospedados gratuitamente.

1.8.7 Portal de qualidade para cadeias fonte

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

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



Veja também:

Ficheiro de idioma intermédio, *Ficheiro de idioma base monolingue*, *Formatos bilíngues e monolíngues*

1.8.8 Revisões de cadeias fonte

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

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

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

Veja também:

Formatos bilíngues e monolíngues, *Revisores dedicados*, labels, *Comentários*

1.9 Perguntas frequentes

1.9.1 Configuração

Como criar um fluxo de trabalho automatizado?

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

1. Configure seu repositório Git para informar Weblate quando houver alguma alteração, veja *Hooks de notificação* para informações sobre como fazer isso.

2. Defina uma URL de push na sua *Configuração de componente* no Weblate, isto permite que o Weblate faça o push das alterações para o seu repositório.
3. Ative o *Enviar ao submeter* na sua *Configuração de componente* no Weblate, isto fará com que o Weblate faça push das alterações no seu repositório sempre que elas acontecerem no Weblate.

Veja também:

Tradução contínua, *Evitar conflitos de mesclagem*

Como acessar os repositórios por SSH?

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

Como resolver conflitos de mesclagem nas traduções?

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

Se já encontrou um conflito de mesclagem, a maneira mais fácil de resolver todos os conflitos localmente no seu sistema é adicionar o Weblate como um repositório remoto, mesclá-lo com upstream e corrigir quaisquer conflitos. Uma vez que fizer push das mudanças de volta, Weblate será capaz de usar a versão unida sem nenhuma outra ação especial.

Nota: Dependendo da sua configuração, o acesso ao repositório Weblate pode requerer autenticação. Ao utilizar o *Exportador git* embutido no Weblate, você se autentica com seu nome de utilizador e a chave da API.

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

# Update weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

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

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

# Open Weblate for translation:
wlc unlock
```

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

```
# Add and update Weblate remotes
git remote add weblate-one https://hosted.weblate.org/git/project/one/
git remote add weblate-second https://hosted.weblate.org/git/project/second/
git remote update weblate-one weblate-second

# Merge QA_4_7 branch:
git checkout QA_4_7
git merge weblate-one/QA_4_7
... # Resolve conflicts
git commit

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

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

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

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

```
# Add remote:
git remote add weblate /path/to/weblate/snapshot/

# Update Weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

# Resolve conflicts in the PO files:
for PO in `find . -name '*.po'` ; do
    msgcat --use-first /path/to/weblate/snapshot/$PO\
               /path/to/upstream/snapshot/$PO -o $PO.merge
    msgmerge --previous --lang=${PO%.po} $PO.merge domain.pot -o $PO
    rm $PO.merge
    git add $PO
done
git commit

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

Veja também:

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

Como faço para traduzir vários ramos ao mesmo tempo?

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

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

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

Veja também:

Manter traduções iguais entre componentes

Como traduzir projetos multiplataforma?

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

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

Veja também:

Manter traduções iguais entre componentes

Como exportar o repositório Git que a Weblate usa?

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

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

Alternativamente, pode usar *Exportador git* dentro do Weblate para automatizar isto.

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

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

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

Claro que é livre para misturar tudo isso como quiser.

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

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

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

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

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

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

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

Como posso verificar se a minha Weblate está configurada corretamente?

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

Veja também:

Monitorando o Weblate, *Monitorar o estado do Celery*

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

Este é o nome padrão do committer, configurado por `DEFAULT_COMMITER_EMAIL` e `DEFAULT_COMMITER_NAME`.

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

Para commits onde nenhuma autoria é conhecida (por exemplo, sugestões anônimas ou resultados de tradução automática), a autoria é creditada ao utilizador anônimo (veja `ANONYMOUS_USER_NAME`). Pode alterar o nome e o e-mail na interface de gestão.

Veja também:

Configuração de componente

Como mover ficheiros no repositório sem perder o histórico no Weblate?

Para manter o histórico, comentários ou capturas de ecrã vinculados às cadeias após alterar a localização dos ficheiros, precisa garantir que essas cadeias nunca sejam apagadas do Weblate. Essas remoções podem acontecer caso o repositório Weblate seja atualizado, mas a configuração dos componentes ainda aponta aos ficheiros antigos. Isso faz o Weblate assumir que ele deve apagar todas as traduções.

A solução para isso é realizar a operação em sincronia com Weblate:

1. Bloquear o componente afetado na Weblate.
2. Fazer commit de todas as alterações pendentes e mesclá-las no repositório upstream.
3. Desativar o recebimento de webhooks o :ref:"project"; isto impede que o Weblate veja imediatamente alterações no repositório.

4. Faça todas as alterações necessárias no repositório (por exemplo, a usar `git mv`), faça o push delas para o repositório upstream.
5. Altere o :ref:`component" para corresponder à nova configuração; ao alterar a configuração, Weblate irá buscar o repositório atualizado e notará as localizações alteradas enquanto mantém as cadeias existentes.
6. Desbloqueie o componente e reative os ganchos na configuração do projeto.

1.9.2 Uso

Como faço para revisar as traduções dos outros?

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

Veja também:

Fluxos de trabalho de tradução

Como posso fornecer feedback sobre uma cadeia fonte?

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

Veja também:

report-source, *Comentários*

Como posso usar as traduções existentes enquanto traduzo?

- Todas as traduções no Weblate podem ser usadas graças à memória de tradução compartilhada.
- Pode importar ficheiros de memória de tradução existentes para o Weblate.
- Use a funcionalidade de importação para carregar compêndios como traduções, sugestões ou traduções que necessitem de revisão. Esta é a melhor abordagem para uma tradução única a usar um compêndio ou uma base de dados de tradução similar.
- You can set up *tmserver* with all databases you have and let Weblate use it. This is good when you want to use it several times during translation.
- Outra opção é traduzir todos os projetos relacionados numa única instância Weblate, o que o fará pegar automaticamente as traduções de outros projetos também.

Veja também:

Configuring automatic suggestions, *Sugestões automáticas*, *Memória de Tradução*

O Weblate atualiza ficheiros de tradução além de traduções?

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

Veja também:

updating-target-files

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

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

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

Veja também:

Definições de idioma

Weblate pode destacar alterações numa cadeia aproximada?

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

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

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

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

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

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

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

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

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

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

Veja também:

updating-target-files

1.9.3 Soluções de problemas

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

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

A maneira mais fácil de fazer isto é executar:

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

Veja também:

DATA_DIR

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

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

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

Veja também:

Configuração de hosts permitidos

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

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

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

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

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

1.9.4 Funcionalidades

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

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

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

Weblate também tem suporte à operação sem VCS, veja *Ficheiros locais*.

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

Veja também:

Integração de controlo de versões

Como o Weblate credita os tradutores?

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

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

Veja também:

`list_translators`, ../devel/reporting

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

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

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

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

Estes são códigos de idioma definidos por **RFC 5646** 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`.

Nota: O padrão do Weblate é o código de idioma do estilo POSIX com sublinhado, veja *Definições de idioma* para mais detalhes.

Veja também:

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

1.10 Formatos de ficheiros suportados

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

Veja também:

Translation Related File Formats

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

1.10.1 Formatos bilíngues e monolíngues

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

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

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

1.10.2 Deteção automática

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

1.10.3 Capacidades dos tipos de tradução

Capacidades de todos os formatos suportados:

Formato	Linguali-dade[Página 80, 1]	Plu-rais[Página 80, 2]	Descri-ção[Página 80, 3]	Con-texto[Página 80, 4]	Localiza-ção[Página 80, 5]	Marcado-res[Página 80, 8]	Estados adicio-nais[Página 80, 6]
GNU gettext	bilíngue	sim	sim	sim	sim	sim[9]	necessita edição
Gettext mo-nolíngue	mono	sim	sim	sim	sim	sim[9]	necessita edição
XLIFF	ambos	sim	sim	sim	sim	sim[10]	necessita edição, aprovado
Propri-edades Java	ambos	não	sim	não	não	não	
Ficheiros lang mi18n	mono	não	sim	não	não	não	

continues on next page

Table 1 – continuação da página anterior

Formato	Linguali-dade[Página 80, 1]	Plurais[2]	Descri-ção[3]	Contexto[4]	Localiza-ção[5]	Marcado-res[8]	Estados adicio-nais[6]
Propri-edades GWT	mono	sim	sim	não	não	não	
Tradu-ções para Joomla	mono	não	sim	não	sim	não	
.ts de Qt Linguist	ambos	sim	sim	não	sim	sim[Página 80, 10]	necessita edição
Recurso de cadeias de Android	mono	sim	sim[7]	não	não	sim[Página 80, 10]	
Cadeias de Apple iOS	ambos	não	sim	não	não	não	
Cadeias de PHP	mono	não[11]	sim	não	não	não	
Ficheiro JSON	mono	não	não	não	não	não	
Ficheiros i18next do JSON	mono	sim	não	não	não	não	
Ficheiros JSON go-i18n	mono	sim	não	não	não	não	
Ficheiro ARB	mono	sim	sim	não	não	não	
JSON We-bExtension	mono	sim	sim	não	não	não	
Ficheiros de recursos .XML	mono	não	sim	não	não	sim[Página 80, 10]	
Resource-Dictionary files	mono	não	não	não	não	sim[Página 80, 10]	
Ficheiros CSV	ambos	não	sim	sim	sim	não	necessita edição
Ficheiros YAML	mono	não	sim	não	não	não	
Fichei-ros Ruby YAML	mono	sim	sim	não	não	não	
Ficheiros DTD	mono	não	não	não	não	não	
Ficheiro XML simples	mono	não	não	não	não	sim[Página 80, 10]	
Ficheiros RC do Windows	mono	não	sim	não	não	não	
Excel Open XML	mono	não	sim	sim	sim	não	necessita edição

continues on next page

Table 1 – continuação da página anterior

Formato	Linguali-dade[Página 80, 1]	Plurais[2]	Descri-ção[3]	Contexto[4]	Localiza-ção[5]	Marcado-res[8]	Estados adicio-nais[6]
Ficheiros de metada-dos da App Store	mono	não	não	não	não	não	
Ficheiros de legenda	mono	não	não	não	sim	não	
Ficheiro HTML	mono	não	não	não	não	não	
Formato OpenDo-cument	mono	não	não	não	não	não	
Formato IDML	mono	não	não	não	não	não	
Traduções de INI	mono	não	não	não	não	não	
Traduções Inno Setup INI	mono	não	não	não	não	não	
Formato TermBase eXchange	bilíngue	não	sim	não	não	sim[Página 80, 10]	
Ficheiros de texto	mono	não	não	não	não	não	
Formato stringsdict	mono	sim	sim	não	não	não	
Formato Fluent	mono	não[12]	sim	não	não	não	

Textos somente leitura

Novo na versão 3.10.

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

[1] Veja *Formatos bilíngues e monolíngues*

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

[3] As descrições da cadeia fonte podem ser usadas para passar informações adicionais sobre a cadeia a traduzir.

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

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

[8] Veja *Personalizando comportamento a usar marcadores*

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

[9] Os comentários do tipo gettext são usados como marcadores.

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

[7] Comentário XML posto antes do elemento `<string>`, analisado como uma descrição da cadeia fonte.

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

[12] Os plurais são tratados na sintaxe dos textos e não são expostos como plurais no Weblate.

1.10.4 GNU gettext

O formato mais usado para traduzir software livre.

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

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

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

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

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`po/*.po`
Ficheiro de idioma base monolingue	*Vazio*
Modelo para novas traduções	`po/messages.pot`
Formato de ficheiro	*Ficheiro Gettext PO*

Veja também:

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

Gettext monolíngue

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

O ficheiro gettext PO monolíngue normalmente se parece com este:

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

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

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

Enquanto o ficheiro de idioma base será:

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

(continues on next page)

(continuação da página anterior)

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

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

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

1.10.5 XLIFF

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

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

Weblate supports XLIFF in several variants:

XLIFF translation file
Simple XLIFF file where content of the elements is stored as plain text (all XML elements being escaped).

XLIFF with placeables support
Standard XLIFF supporting placeables and other XML elements.

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

Veja também:

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

Estados de tradução

Alterado na versão 3.3: Weblate ignorava o atributo `state` antes da versão 3.3.

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

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

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

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

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

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

Veja também:

Revisores dedicados

Espaços em branco e novas linhas em XLIFF

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

Por exemplo:

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

Especificação de bandeiras de tradução

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

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

O atributo `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 *Gerir letras*).

Chaves de cadeias

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

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

Configuração de componente típica do Weblate para XLIFF bilíngue	
Máscara de ficheiros	`localizations/*.xliff`
Ficheiro de idioma base monolingue	*Vazio*
Modelo para novas traduções	`localizations/en-US.xliff`
Formato de ficheiro	*Ficheiro de tradução XLIFF*

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

Veja também:

XLIFF na Wikipédia, XLIFF, Atributo font no XLIFF 1.2, Atributo maxwidth no XLIFF 1.2

1.10.6 Propriedades Java

Formato nativo do Java para traduções.

As propriedades Java são geralmente usadas como traduções monolíngues.

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

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

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

Veja também:

Java properties na Wikipédia, Mozilla and Java properties files, *Ficheiros lang mi18n*, *Propriedades GWT*, updating-target-files, *Format the Java properties file*, *Limpeza de ficheiros de tradução*

1.10.7 Ficheiros lang mi18n

Novo na versão 4.7.

Formato de ficheiro usado para localização de JavaScript por mi18n. Sintaticamente, ele corresponde a *Propriedades Java*.

Configuração de componente típica do Weblate	
Máscara de ficheiros	`*.lang`
Ficheiro de idioma base monolingue	`en-US.lang`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro lang mi18n*

Veja também:

mi18n, Mozilla and Java properties files, *Propriedades Java*, updating-target-files, *Format the Java properties file*, *Limpeza de ficheiros de tradução*

1.10.8 Propriedades GWT

Formato nativo de GWT para traduções.

As propriedades GWT são geralmente usadas como traduções monolíngues.

Configuração de componente típica do Weblate	
Máscara de ficheiros	`src/app/Bundle_*.properties`
Ficheiro de idioma base monolingue	`src/app/Bundle.properties`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Propriedades GWT*

Veja também:

GWT localization guide, GWT Internationalization Tutorial, Mozilla and Java properties files, updating-target-files, *Format the Java properties file*, *Limpeza de ficheiros de tradução*

1.10.9 Traduções de INI

Novo na versão 4.1.

Formato de aplicação INI para traduções.

As traduções de INI são geralmente usadas como traduções monolíngues.

Configuração de componente típica do Weblate	
Máscara de ficheiros	`language/*.ini`
Ficheiro de idioma base monolingue	`language/en.ini`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro INI*

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

Veja também:

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

1.10.10 Traduções Inno Setup INI

Novo na versão 4.1.

Formato de ficheirio Inno Setup INI para traduções.

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

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`language/*.islu`
Ficheiro de idioma base monolingue	`language/en.islu`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro Inno Setup INI*

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

Veja também:

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

1.10.11 Traduções para Joomla

Novo na versão 2.12.

Formato nativo de Joomla para traduções.

As traduções de Joomla são geralmente usadas como traduções monolíngues.

Configuração de componente típica do Weblate	
Máscara de ficheiros	`language/*/com_foobar.ini`
Ficheiro de idioma base monolingue	`language/en-GB/com_foobar.ini`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro de idioma Joomla*

Veja também:

Mozilla and Java properties files, *Traduções de INI*, *Traduções Inno Setup INI*

1.10.12 .ts de Qt Linguist

Formato de tradução usado em aplicações baseados em Qt.

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

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

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

Veja também:

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

1.10.13 Recurso de cadeias de Android

Formato de ficheiro específico do Android para traduzir aplicações.

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

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

Veja também:

Documentação de recursos de cadeias de Android, Android string resources

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

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

se torna:

```
<string-array name="several_strings">
    <item>@string/several_strings_0</item>
    <item>@string/several_strings_1</item>
</string-array>
<string name="several_strings_0">First string</string>
<string name="several_strings_1">Second string</string>
```

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

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

1.10.14 Cadeias de Apple iOS

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

Apple iOS strings are usually used as monolingual.

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

Veja também:

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

1.10.15 Cadeias de PHP

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

Exemplo de ficheiro:

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

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

Cadeias de PHP Laravel

Alterado na versão 4.1.

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

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

Veja também:

PHP, Localização de Laravel

1.10.16 Ficheiro JSON

Novo na versão 2.0.

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

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

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

O Weblate atualmente suporta várias variantes das traduções JSON:

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

- *JSON WebExtension*
- *Ficheiro ARB*

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

Exemplo de ficheiro:

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

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

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

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

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

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

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

Veja também:

JSON, updating-target-files, *Personalizar a saída JSON*, *Limpeza de ficheiros de tradução*,

1.10.17 Ficheiros i18next do JSON

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

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

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

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

A variante v4 usa uma abordagem diferente para armazenar plurais e atualmente não é compatível.

Exemplo de ficheiro:

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

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

Veja também:

JSON, JSON Format de i18next, updating-target-files, *Personalizar a saída JSON*, *Limpeza de ficheiros de tradução*

1.10.18 Ficheiros JSON go-i18n

Novo na versão 4.1.

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

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`langs/*.json`
Ficheiro de idioma base monolingue	`langs/en.json`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*ficheiro JSON go-i18n*

Veja também:

JSON, go-i18n, updating-target-files, *Personalizar a saída JSON*, *Limpeza de ficheiros de tradução*,

1.10.19 Ficheiro ARB

Novo na versão 4.1.

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`lib/l10n/intl_*.arb`
Ficheiro de idioma base monolingue	`lib/l10n/intl_en.arb`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro ARB*

Veja também:

JSON, Application Resource Bundle Specification, Internationalizing Flutter apps, updating-target-files, *Personalizar a saída JSON*, *Limpeza de ficheiros de tradução*

1.10.20 JSON WebExtension

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

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

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

Exemplo de ficheiro:

```
{
  "hello": {
    "message": "Ahoj světe!\n",
    "description": "Description",
    "placeholders": {
      "url": {
        "content": "$1",
        "example": "https://developer.mozilla.org"
      }
    }
  },
  "orangutan": {
    "message": "Orangutan has $coUnT$ bananas",
    "description": "Description",
    "placeholders": {
      "count": {
        "content": "$1",
        "example": "5"
      }
    }
  },
  "try": {
    "message": "",
    "description": "Description"
  },
  "thanks": {
    "message": "",
    "description": "Description"
  }
}
```

Configuração de componente típica do Weblate	
Máscara de ficheiros	`_locales/*/messages.json`
Ficheiro de idioma base monolingue	`_locales/en/messages.json`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro JSON WebExtension*

Veja também:

JSON, Google chrome.i18n, Mozilla Extensions Internationalization

1.10.21 Ficheiros de recursos .XML

Novo na versão 2.3.

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`Resources/Language.*.resx`
Ficheiro de idioma base monolingue	`Resources/Language.resx`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro de recursos .NET*

Veja também:

.NET Resource files (.resx), updating-target-files, *Limpeza de ficheiros de tradução*

1.10.22 ResourceDictionary files

Novo na versão 4.13.

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`Languages/*.xaml`
Ficheiro de idioma base monolingue	`Language/en.xaml`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*ResourceDictionary file*

Veja também:

Flat XML, *Ficheiro XML simples*, updating-target-files, *Limpeza de ficheiros de tradução*

1.10.23 Ficheiros CSV

Novo na versão 2.4.

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

- Ficheiros com campos de definição de cabeçalho (`location`, `source`, `target`, `ID`, `fuzzy`, `context`, `translator_comments`, `developer_comments`). Esta é a abordagem recomendada, pois é a menos sujeita a erros. Escolha *Ficheiro CSV* como um formato de ficheiro.
- Ficheiros com dois campos – fonte e tradução (nesta ordem). Escolha *Ficheiro CSV simples* como um formato de ficheiro.

- Ficheiros sem cabeçalhos com campos na ordem definida pelo translate-toolkit: `location`, `source`, `target`, `ID`, `fuzzy`, `context`, `translator_comments`, `developer_comments`. Escolha *Arquivo CSV* como formato de ficheiro.
- Lembre-se de definir o *Ficheiro de idioma base monolingue* quando os seus ficheiros são monolíngues (veja *Formatos bilíngues e monolíngues*).

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

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

Exemplo de ficheiro:

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

Configuração de componente típica do Weblate para CSV bilíngue	
Máscara de ficheiros	`locale/*.csv`
Ficheiro de idioma base monolingue	*Vazio*
Modelo para novas traduções	`locale/en.csv`
Formato de ficheiro	*Ficheiro CSV*

Configuração de componente típica do Weblate para CSV monolíngue	
Máscara de ficheiros	`locale/*.csv`
Ficheiro de idioma base monolingue	`locale/en.csv`
Modelo para novas traduções	`locale/en.csv`
Formato de ficheiro	*Ficheiro CSV simples*

Multivalue CSV file

Novo na versão 4.13.

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

Veja também:

CSV

1.10.24 Ficheiros YAML

Novo na versão 2.9.

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

Exemplo de um ficheiro YAML:

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`translations/messages.*.yml`
Ficheiro de idioma base monolingue	`translations/messages.en.yml`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro YAML*

Veja também:

YAML, *Ficheiros Ruby YAML*

1.10.25 Ficheiros Ruby YAML

Novo na versão 2.9.

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

Exemplo de ficheiro YAML de i18n do Ruby:

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`translations/messages.*.yml`
Ficheiro de idioma base monolingue	`translations/messages.en.yml`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro Ruby YAML*

Veja também:

YAML, *Ficheiros YAML*

1.10.26 Ficheiros DTD

Novo na versão 2.18.

Exemplo de ficheiro DTD:

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`locale/*.dtd`
Ficheiro de idioma base monolingue	`locale/en.dtd`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro DTD*

Veja também:

Mozilla DTD format

1.10.27 Ficheiro XML simples

Novo na versão 3.9.

Exemplo de um ficheiro XML simples:

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`locale/*.xml`
Ficheiro de idioma base monolingue	`locale/en.xml`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro XML simples*

Veja também:

Flat XML

1.10.28 Ficheiros RC do Windows

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

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

Exemplo de ficheiro RC do Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

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

Configuração de componente típica do Weblate	
Máscara de ficheiros	`lang/*.rc`
Ficheiro de idioma base monolingue	`lang/en-US.rc`
Modelo para novas traduções	`lang/en-US.rc`
Formato de ficheiro	*Ficheiro RC*

Veja também:

Windows RC files

1.10.29 Ficheiros de metadados da App Store

Novo na versão 3.5.

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

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

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

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

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

1.10.30 Ficheiros de legenda

Novo na versão 3.7.

Weblate pode traduzir vários ficheiros de legenda:

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

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

Veja também:

Subtitles

1.10.31 Excel Open XML

Novo na versão 3.2.

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

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

1.10.32 Ficheiro HTML

Novo na versão 4.1.

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

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

Veja também:

HTML

1.10.33 Ficheiros de texto

Novo na versão 4.6.

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

O conteúdo traduzível é extraído dos ficheiros de texto simples e oferecido para a tradução. Cada parágrafo é traduzido como um texto separado.

Há três sabores desse formato:

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

Veja também:

Simple Text Documents

1.10.34 Formato OpenDocument

Novo na versão 4.1.

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

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

Veja também:

OpenDocument Format

1.10.35 Formato IDML

Novo na versão 4.1.

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

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

1.10.36 Formato TermBase eXchange

Novo na versão 4.5.

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

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

Veja também:

TBX na Wikipédia, TBX, *Glossário*

1.10.37 Formato stringsdict

Novo na versão 4.8.

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

Formato baseado em XML usado pela Apple, que é capaz de armazenar formas plurais de um texto.

Configuração de componente típica do Weblate	
Máscara de ficheiros	`Resources/*.lproj/Localizable.stringsdict`
Ficheiro de idioma base monolingue	`Resources/en.lproj/Localizable.stringsdict` ou `Resources/Base.lproj/Localizable.stringsdict`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro Stringsdict*

Veja também:

Cadeias de Apple iOS, Formato de ficheiro Stringsdict

1.10.38 Formato Fluent

Novo na versão 4.8.

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

Fluent é um formato de texto monolíngue que se concentra na localização assimétrica: um texto simples num idioma pode ser mapeado para uma tradução multivariante complexa em outro idioma.

Configuração de componente típica do Weblate	
Máscara de ficheiros	`locales/*/messages.ftl`
Ficheiro de idioma base monolingue	`locales/en/messages.ftl`
Modelo para novas traduções	*Vazio*
Formato de ficheiro	*Ficheiro Fluent*

Veja também:

Site do projeto Fluent

1.10.39 Suportando outros formatos

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.

Para adicionar suporte para um novo formato, a abordagem preferida é primeiro implementar o suporte para ele no translate-toolkit.

Veja também:

Translation Related File Formats

1.11 Integração de controlo de versões

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

1.11.1 Acessando repositórios

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

Acessando repositórios do Hosted Weblate

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

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

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

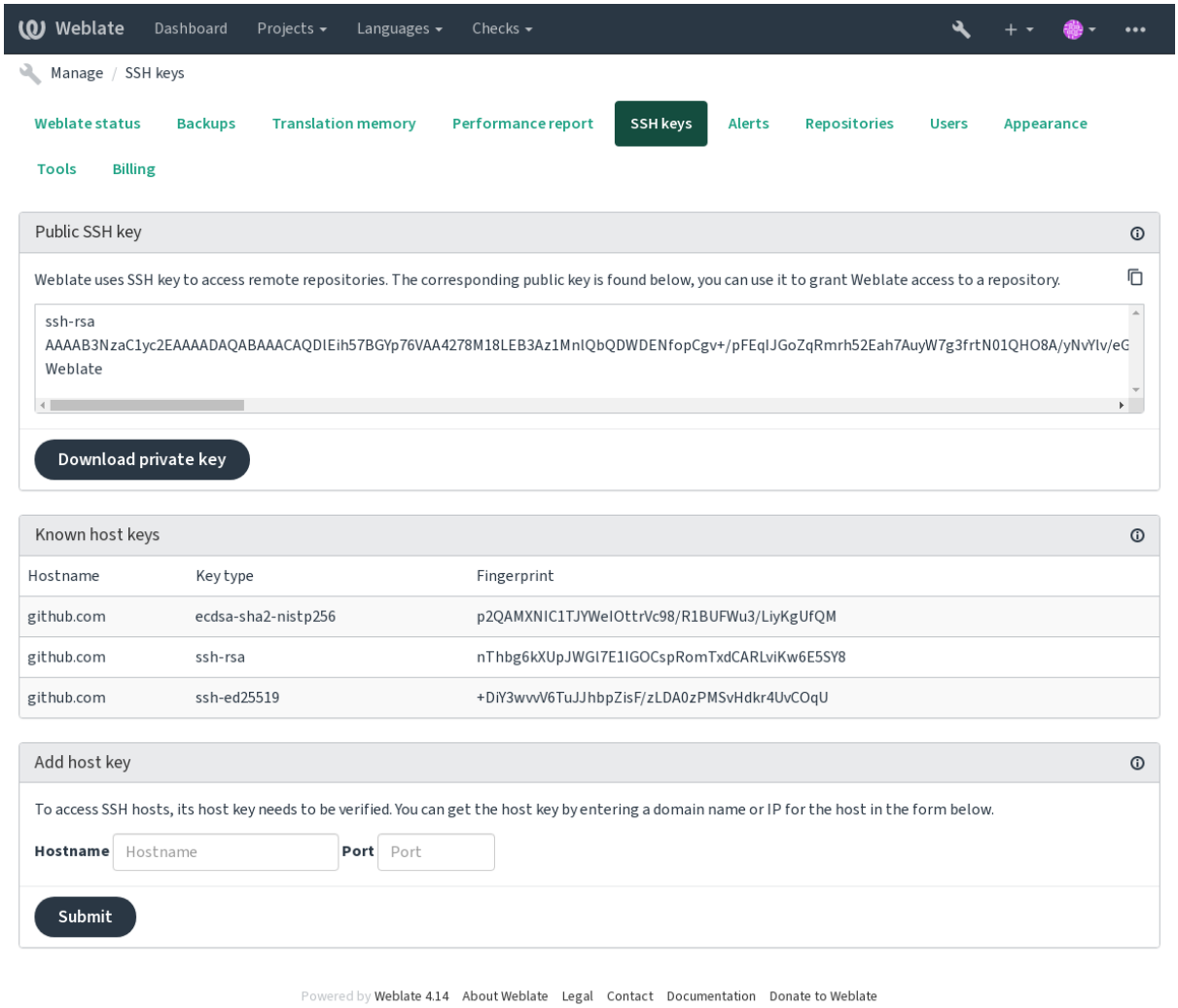
Repositórios SSH

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

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

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

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



Chave SSH do Weblate

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

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

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

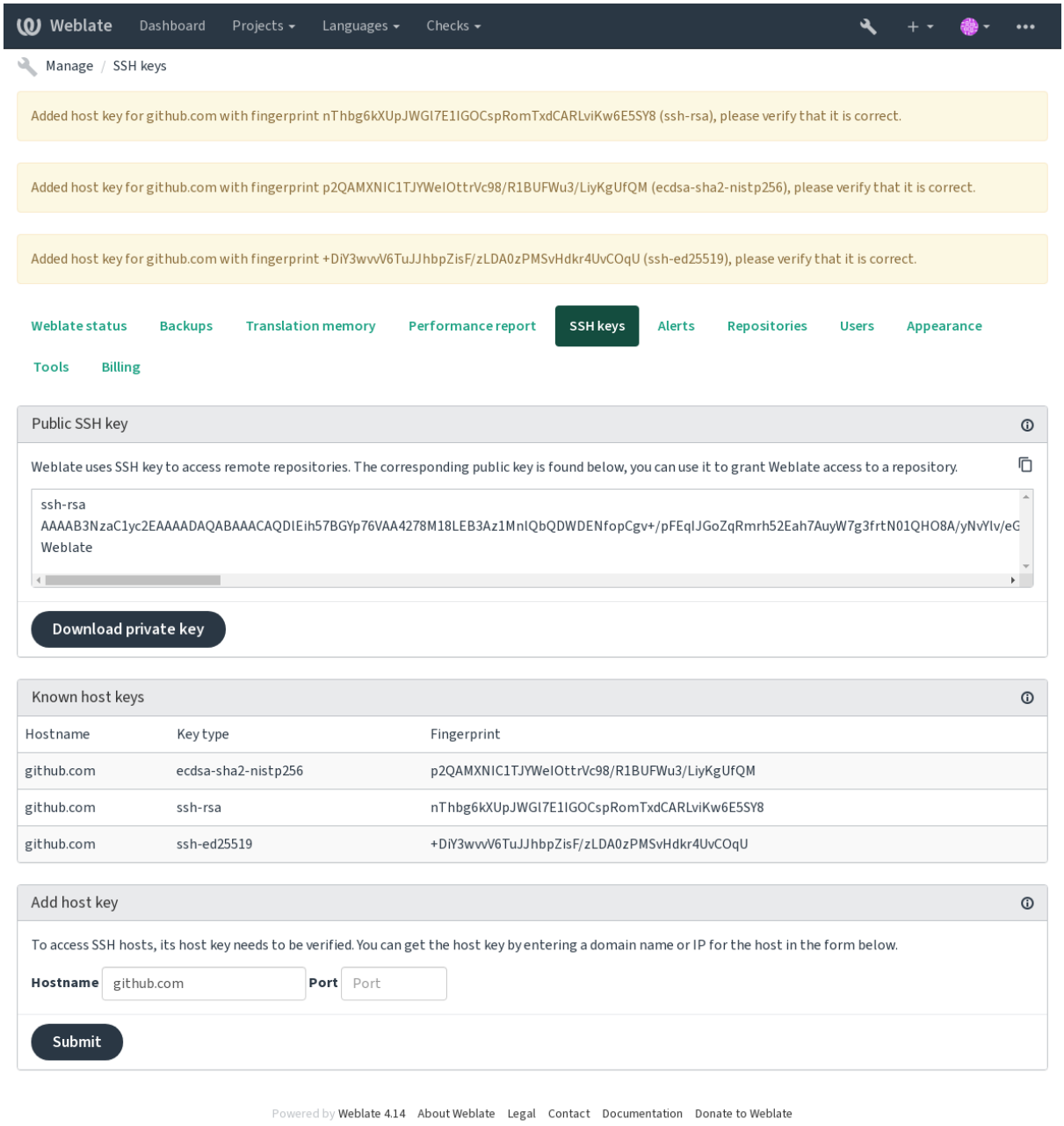
Dica: Faça um backup da chave SSH privada gerada do Weblate.

Verificando chaves SSH do host

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

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

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



Repositórios do GitHub

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

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

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

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

Veja também:

Acessando repositórios do Hosted Weblate

URLs internas do Weblate

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

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

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

Motivos para usar isso:

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

Repositórios HTTPS

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

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

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

Usando proxy

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

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

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

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

Veja também:

A página man do cURL, Documentação do git config

1.11.2 Git

Dica: O Weblate requer o Git 2.12 ou mais recente.

Veja também:

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

Git com push forçado

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

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

Personalizando a configuração do Git

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

Auxiliares de remotos do Git

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

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

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

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

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

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

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

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

1.11.3 GitHub pull requests

Novo na versão 2.3.

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

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

Precisa configurar as credenciais da API (`GITHUB_CREDENTIALS`) nas configurações do Weblate par fazê-lo funcionar. Uma vez configurado, verá uma opção *GitHub* ao selecionar *Sistema de controlo de versões*.

Veja também:

Fazendo push das alterações do Weblate, `GITHUB_USERNAME`, `GITHUB_TOKEN`, `GITHUB_CREDENTIALS`

1.11.4 Solicitação de mesclagem do GitLab

Novo na versão 3.9.

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

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

Precisa configurar as credenciais da API (`GITLAB_CREDENTIALS`) nas configurações da Weblate para funcionar. Uma vez configurado, verá uma opção *GitLab* ao selecionar *Sistema de controlo de versões*.

Veja também:

Fazendo push das alterações do Weblate, `GITLAB_USERNAME`, `GITLAB_TOKEN`, `GITLAB_CREDENTIALS`

1.11.5 Gitea pull requests

Novo na versão 4.12.

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

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

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

Veja também:

Fazendo push das alterações do Weblate, `GITEA_USERNAME`, `GITEA_TOKEN`, `GITEA_CREDENTIALS`

1.11.6 Pagure merge requests

Novo na versão 4.3.2.

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

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

Precisa configurar as credenciais da API (`PAGURE_CREDENTIALS`) nas configurações da Weblate para funcionar. Uma vez configurado, verá uma opção *Pagure* ao selecionar *Sistema de controlo de versões*.

Veja também:

Fazendo push das alterações do Weblate, `PAGURE_USERNAME`, `PAGURE_TOKEN`, `PAGURE_CREDENTIALS`

1.11.7 Gerrit

Novo na versão 2.2.

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

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

1.11.8 Mercurial

Novo na versão 2.1.

Mercurial é outro VCS que pode usar diretamente no Weblate.

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

Veja também:

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

1.11.9 Subversion

Novo na versão 2.8.

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

Nota: O Weblate tenta detetar o layout do repositório Subversion automaticamente – ele tem suporta a URLs diretas para remos ou repositórios com layout padrão (branches/, tags/ e trunk/). Mais informações sobre isso podem ser encontradas na documentação do git-svn. Se o repositório não tiver um layout padrão e encontrar erros, tente incluir o nome do ramo na URL do repositório e deixar a ramo vazia.

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

Credenciais de Subversion

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

```
# Use DATA_DIR as configured in Weblate settings.py, it is /app/data in the Docker
HOME=${DATA_DIR}/home svn co https://svn.example.com/example
```

Veja também:

`DATA_DIR`

1.11.10 Ficheiros locais

1.11.11 Git

Dica: Internamente, ele usa *Git*. Ele requer Git instalado e permite que mude para usar o Git nativamente com histórico completo das suas traduções.

Novo na versão 3.8.

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

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

1.12 API REST do Weblate

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

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

1.12.1 Autenticação e parâmetros genéricos

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

ANY /

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

Parâmetros de consulta

- **`format`** – Formato de resposta (substitui Accept). Os valores possíveis dependem da configuração do quadro REST, por padrão `json` e `api` são suportados. Este último fornece uma interface de navegador web para API.
- **`page`** – Retorna determinada página de resultados paginados (use os campos *next* e *previous* em resposta para automatizar a navegação).

Cabeçalhos de solicitação

- Accept – o tipo de conteúdo de resposta depende do cabeçalho Accept
- Authorization – token opcional para se autenticar como `Authorization: Token SEU-TOKEN`

Cabeçalhos de resposta

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

Objeto JSON de resposta

- **`detail`** (*`string`*) – descrição detalhada do resultado (para códigos de estado HTTP diferentes de 200 OK)
- **`count`** (*`int`*) – contagem total de itens para listas de objetos

- **next** (*string*) – URL da próxima página para listas de objetos
- **previous** (*string*) – URL da página anterior para listas de objetos
- **results** (*array*) – resultados para listas de objetos
- **url** (*string*) – URL para acessar esse recurso a usar API
- **web_url** (*string*) – URL para acessar esse recurso a usar navegador web

Códigos de estado

- 200 OK – quando o pedido foi resolvido corretamente
- 201 Created – quando um novo objeto foi criado com sucesso
- 204 No Content – quando um objeto foi excluído com sucesso
- 400 Bad Request – quando faltam os parâmetros do formulário
- 403 Forbidden – quando o acesso é negado
- 429 Too Many Requests – quando a limitação está em vigor

Tokens de autenticação

Alterado na versão 4.10: Tokens com escopo de projeto foram introduzidos na versão 4.10.

Cada utilizador tem o seu próprio código de acesso que pode ser obtido no perfil de utilizador. Os tokens de utilizadores recém-gerados têm o prefixo `wlu_`.

É possível criar tokens com escopo de projeto para acesso API apenas a um determinado projeto. Estes tokens podem ser identificados pelo prefixo `wlp_`.

Exemplos de autenticação

Exemplo de solicitação:

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

Exemplo de resposta:

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

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

Exemplo de CURL:

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

Exemplos de passagem de parâmetros

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

Exemplo de solicitação de formulário:

```
POST /api/projects/hello/repository/ HTTP/1.1
Host: example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Token TOKEN

operation=pull
```

Exemplo de solicitação JSON:

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

{"operation":"pull"}
```

Exemplo de CURL:

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

Exemplo de CURL JSON:

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

Limitação de taxa da API

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

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

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

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

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

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

Veja também:

Limitação de taxa, *Limitação de taxa*, *WEBLATE_API_RATELIMIT_ANON*, *WEBLATE_API_RATELIMIT_USER*

1.12.2 Ponto de entrada da API

GET /api/

O ponto de entrada raiz da API.

Exemplo de solicitação:

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

Exemplo de resposta:

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

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

1.12.3 Utilizadores

Novo na versão 4.0.

GET /api/users/

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

Veja também:

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

POST /api/users/

Cria um novo utilizador.

Parâmetros

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

- **email** (*string*) – E-mail do utilizador
- **is_superuser** (*boolean*) – O utilizador é superutilizador? (opcional)
- **is_active** (*boolean*) – O utilizador está ativo? (opcional)
- **is_bot** (*boolean*) – O utilizador é bot? (opcional) (usado para tokens no escopo do projeto)

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

Retorna informações sobre os utilizadores.

Parâmetros

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

Objeto JSON de resposta

- **username** (*string*) – nome de utilizador de um utilizador
- **full_name** (*string*) – nome completo de um utilizador
- **email** (*string*) – email de um utilizador
- **is_superuser** (*boolean*) – se o utilizador é um superutilizador
- **is_active** (*boolean*) – se o utilizador está ativo
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – data de criação do utilizador
- **groups** (*array*) – ligações para grupos associados; veja *GET /api/groups/(int:id)/*

Exemplo de dados JSON:

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

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

Altera os parâmetros do utilizador.

Parâmetros

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

Objeto JSON de resposta

- **username** (*string*) – nome de utilizador de um utilizador
- **full_name** (*string*) – nome completo de um utilizador
- **email** (*string*) – email de um utilizador
- **is_superuser** (*boolean*) – se o utilizador é um superutilizador

- **is_active** (*boolean*) – se o utilizador está ativo
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – data de criação do utilizador

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

Altera os parâmetros do utilizador.

Parâmetros

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

Objeto JSON de resposta

- **username** (*string*) – nome de utilizador de um utilizador
- **full_name** (*string*) – nome completo de um utilizador
- **email** (*string*) – email de um utilizador
- **is_superuser** (*boolean*) – se o utilizador é um superutilizador
- **is_active** (*boolean*) – se o utilizador está ativo
- **is_bot** (*boolean*) – whether the user is bot (used for project scoped tokens)
- **date_joined** (*string*) – data de criação do utilizador

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

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

Parâmetros

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

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

Associa grupos a um utilizador.

Parâmetros

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

Parâmetros de formulário

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

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

Novo na versão 4.13.1.

Remove user from a group.

Parâmetros

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

Parâmetros de formulário

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

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

Lista as estatísticas de um utilizador.

Parâmetros

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

Objeto JSON de resposta

- **translated** (*int*) – Quantidade de traduções por utilizador
- **suggested** (*int*) – Quantidade de sugestões por utilizador
- **uploaded** (*int*) – Quantidade de envios por utilizador

- **commented** (*int*) – Quantidade de comentários por utilizador
- **languages** (*int*) – Quantidade de idiomas que o utilizador pode traduzir

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

Lista de subscrições de um utilizador.

Parâmetros

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

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

Associar subscrições a um utilizador.

Parâmetros

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

Objeto JSON da solicitação

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

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

Obtém uma assinatura associada a um utilizador.

Parâmetros

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

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

Edita uma assinatura associada a um utilizador.

Parâmetros

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

Objeto JSON da solicitação

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

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

Edita uma assinatura associada a um utilizador.

Parâmetros

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

Objeto JSON da solicitação

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

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

Apaga uma assinatura associada a um utilizador.

Parâmetros

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

1.12.4 Grupos

Novo na versão 4.0.

GET /api/groups/

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

Veja também:

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

POST /api/groups/

Cria um novo grupo.

Parâmetros

- **name** (*string*) – Nome do grupo
- **project_selection** (*int*) – Grupo de seleção de projetos a partir de opções dadas
- **language_selection** (*int*) – Grupo de idiomas selecionados a partir de opções dadas
- **defining_project** (*str*) – link to the defining project, used for *A gerir controle de acesso por projeto*; see *GET /api/projects/(string:project)/*

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

Devolve informação sobre o grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

- **name** (*string*) – nome de um grupo
- **project_selection** (*int*) – inteiro correspondente ao grupo de projetos
- **language_selection** (*int*) – inteiro correspondente ao grupo de idiomas
- **roles** (*array*) – ligacão para funções associadas; veja *GET /api/roles/(int:id)/*
- **projects** (*array*) – ligação para projetos associados; veja *GET /api/projects/(string:project)/*
- **components** (*array*) – ligação para componentes associados; veja *GET /api/components/(string:project)/(string:component)/*
- **componentlists** (*array*) – ligação para lista de componentes associada; veja *GET /api/component-lists/(str:slug)/*
- **defining_project** (*str*) – link to the defining project, used for *A gerir controle de acesso por projeto*; see *GET /api/projects/(string:project)/*

Exemplo de dados JSON:

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

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

Altera os parâmetros do grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

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

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

Altera os parâmetros do grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

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

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

Apaga o grupo.

Parâmetros

- **id** (*int*) – ID do grupo

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

Associar funções a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

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

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

Associa componentes a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

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

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

Eliminar componente de um grupo.

Parâmetros

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

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

Associa projetos a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

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

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

Eliminar projeto de um grupo.

Parâmetros

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

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

Associar línguas a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

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

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

Eliminar língua de um grupo.

Parâmetros

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

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

Associa listas de componentes a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

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

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

Apaga a lista de componentes de um grupo.

Parâmetros

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

1.12.5 Funções

GET /api/roles/

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

Veja também:

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

POST /api/roles/

Cria uma nova função.

Parâmetros

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

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

Retorna informações sobre uma função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

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

Exemplo de dados JSON:

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

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

Altera os parâmetros da função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

- **name** (*string*) – Nome da função

- **permissions** (*array*) – lista de codinomes de permissões

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

Altera os parâmetros da função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

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

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

Apaga a função.

Parâmetros

- **id** (*int*) – ID da função

1.12.6 Idiomas

GET /api/languages/

Retorna uma lista de todos os idiomas.

Veja também:

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

POST /api/languages/

Cria uma nova língua.

Parâmetros

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

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

Devolve informação sobre uma língua.

Parâmetros

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

Objeto JSON de resposta

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

Objeto JSON da solicitação

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

Exemplo de dados JSON:

```
{
    "code": "en",
    "direction": "ltr",
    "name": "English",
    "population": 159034349015,
    "plural": {
        "id": 75,
        "source": 0,
        "number": 2,
        "formula": "n != 1",
        "type": 1
    },
    "aliases": [
        "english",
        "en_en",
        "base",
        "source",
        "eng"
    ],
    "url": "http://example.com/api/languages/en/",
    "web_url": "http://example.com/languages/en/",
    "statistics_url": "http://example.com/api/languages/en/statistics/"
}
```

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

Altera os parâmetros linguísticos.

Parâmetros

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

Objeto JSON da solicitação

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

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

Altera os parâmetros linguísticos.

Parâmetros

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

Objeto JSON da solicitação

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

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

Apaga o idioma.

Parâmetros

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

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

Devolve estatísticas para uma língua.

Parâmetros

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

Objeto JSON de resposta

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

1.12.7 Projetos

GET /api/projects/

Retorna uma lista de todos os projetos.

Veja também:

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

POST /api/projects/

Novo na versão 3.9.

Cria um novo projeto.

Parâmetros

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

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

Devolve informação sobre um projecto.

Parâmetros

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

Objeto JSON de resposta

- **name** (*string*) – nome do projeto
- **slug** (*string*) – url amigável do projeto
- **web** (*string*) – site do projeto

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

Exemplo de dados JSON:

```
{
    "name": "Hello",
    "slug": "hello",
    "url": "http://example.com/api/projects/hello/",
    "web": "https://weblate.org/",
    "web_url": "http://example.com/projects/hello/"
}
```

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

Novo na versão 4.3.

Edita um projeto por uma solicitação PATCH.

Parâmetros

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

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

Novo na versão 4.3.

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

Parâmetros

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

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

Novo na versão 3.9.

Apaga um projeto.

Parâmetros

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

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

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

Parâmetros

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

Objeto JSON de resposta

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

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

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

Parâmetros

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

Objeto JSON de resposta

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

Exemplo de dados JSON:

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

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

Realiza a operação dada no repositório VCS.

Parâmetros

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

Objeto JSON da solicitação

- **operation** (*string*) – Operação para realizar: uma de `push`, `pull`, `commit`, `reset`, `cleanup`, `file-sync`

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{"operation":"pull"}
```

Exemplo de resposta JSON:

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

(continues on next page)

(continuação da página anterior)

```
Vary: Accept, Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
Content-Type: application/json
Content-Language: en
Allow: GET, POST, HEAD, OPTIONS

{"result":true}
```

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

Retorna uma lista de componentes de tradução no projeto dado.

Parâmetros

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos componentes; veja *GET /api/components/(string:project)/(string:component)/*

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

Novo na versão 3.9.

Alterado na versão 4.3: Os parâmetros `` zipfile`` e `` docfile`` agora são aceitos para componentes sem VCS, veja *Ficheiros locais*.

Alterado na versão 4.6: Os repositórios clonados agora são automaticamente compartilhados dentro de um projeto a usar *URLs internas do Weblate*. Use `disable_autoshare` para desativar isso.

Cria componentes de tradução no projeto dado.

Dica: Use *URLs internas do Weblate* ao criar vários componentes a partir de um único repositório VCS.

Nota: A maior parte da criação de componentes acontece em segundo plano. Verifique o atributo `task_url` do componente criado e siga o progresso por lá.

Parâmetros

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

Parâmetros de formulário

- **file zipfile** – Ficheiro ZIP para enviar ao Weblate para inicialização de traduções
- **file docfile** – Documento para traduzir
- **boolean disable_autoshare** – Desativa o compartilhamento automático do repositório via *URLs internas do Weblate*.

Objeto JSON da solicitação

- **object** – Parâmetros do componente; veja *GET /api/components/(string:project)/(string:component)/*

Objeto JSON de resposta

- **result** (*object*) – Objeto componente criado; veja *GET /api/components/(string:project)/(string:component)/*

JSON não pode ser usado ao enviar os ficheiros a usar os parâmetros `zipfile` e `docfile`. Os dados devem ser enviados como *multipart/form-data*.

Exemplo de solicitação de formulário com CURL:

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

Exemplo de solicitação JSON com CURL:

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

Solicitação JSON para criar um componente a partir do Git:

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

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

Solicitação JSON para criar um componente a partir de outro:

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

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

(continues on next page)

(continuação da página anterior)

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

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

Retorna estatísticas paginadas para todos os idiomas num projeto.

Novo na versão 3.8.

Parâmetros

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de estatísticas de tradução
- **language** (*string*) – nome de idioma

- **code** (*string*) – código de idioma
- **total** (*int*) – quantidade total de cadeias
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas
- **words_percent** (*float*) – percentagem de palavras traduzidas

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

Retorna estatísticas para um projeto.

Novo na versão 3.8.

Parâmetros

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

Objeto JSON de resposta

- **total** (*int*) – quantidade total de cadeias
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas
- **words_percent** (*float*) – percentagem de palavras traduzidas

1.12.8 Componentes

Dica: Usa `POST /api/projects/(string:project)/repository/` para criar novos componentes.

GET /api/components/

Retorna uma lista de componentes de tradução.

Veja também:

Os atributos do objeto componente estão documentados em `GET /api/components/(string:project)/(string:component)/`.

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

Retorna informações sobre o componente de tradução.

Parâmetros

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

Objeto JSON de resposta

- **project** (*object*) – o projeto de tradução; veja `GET /api/projects/(string:project)/`
- **name** (*string*) – *Nome do componente*
- **slug** (*string*) – *URL amigável do componente*
- **vcs** (*string*) – *Sistema de controlo de versões*

- **repo** (*string*) – *Repositório do código-fonte*
- **git_export** (*string*) – *URL do repositório exportado*
- **branch** (*string*) – *Ramo do repositório*
- **push_branch** (*string*) – *Ramo do push*
- **filemask** (*string*) – *Máscara de ficheiros*
- **template** (*string*) – *Ficheiro de idioma base monolingue*
- **edit_template** (*string*) – *Editar ficheiro base*
- **intermediate** (*string*) – *Ficheiro de idioma intermédio*
- **new_base** (*string*) – *Modelo para novas traduções*
- **file_format** (*string*) – *Formato de ficheiro*
- **license** (*string*) – *Licença da tradução*
- **agreement** (*string*) – *Acordo de contribuidor*
- **new_lang** (*string*) – *Adicionar nova tradução*
- **language_code_style** (*string*) – *Estilo de código de idioma*
- **source_language** (*object*) – objeto do idioma fonte; veja `GET /api/languages/(string:language)/`
- **push** (*string*) – *URL de submissão do repositório*
- **check_flags** (*string*) – *Marcadores de tradução*
- **priority** (*string*) – *Prioridade*
- **enforced_checks** (*string*) – *Verificações impostas*
- **restricted** (*string*) – *Acesso restrito*
- **repoweb** (*string*) – *Navegador do repositório*
- **report_source_bugs** (*string*) – *Endereço para reportar erros na cadeia fonte*
- **merge_style** (*string*) – *Estilo de união*
- **commit_message** (*string*) – *Commit, add, delete, merge, add-on, and merge request messages*
- **add_message** (*string*) – *Commit, add, delete, merge, add-on, and merge request messages*
- **delete_message** (*string*) – *Commit, add, delete, merge, add-on, and merge request messages*
- **merge_message** (*string*) – *Commit, add, delete, merge, add-on, and merge request messages*
- **addon_message** (*string*) – *Commit, add, delete, merge, add-on, and merge request messages*
- **pull_message** (*string*) – *Commit, add, delete, merge, add-on, and merge request messages*
- **allow_translation_propagation** (*string*) – *Permitir propagação da tradução*
- **enable_suggestions** (*string*) – *Ativar sugestões*
- **suggestion_voting** (*string*) – *Votação de sugestão*
- **suggestion_autoaccept** (*string*) – *Aceitar sugestões automaticamente*
- **push_on_commit** (*string*) – *Enviar ao submeter*

- **commit_pending_age** (*string*) – *Idade das alterações a fazer commit*
- **auto_lock_error** (*string*) – *Bloquear com erro*
- **language_regex** (*string*) – *Filtro de idioma*
- **variant_regex** (*string*) – *Expressão regular das variantes*
- **repository_url** (*string*) – URL para estado do repositório; veja `GET /api/components/(string:project)/(string:component)/repository/`
- **translations_url** (*string*) – URL para lista de tradução; veja `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – URL para estado de bloqueio; veja `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – URL para lista de alterações; veja `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – URL para uma tarefa em segundo plano (se houver); veja `GET /api/tasks/(str:uuid)/`

Exemplo de dados JSON:

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

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

Edita uma componente por uma solicitação PATCH.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **source_language** (*string*) – Código de idioma fonte do projeto (opcional)

Objeto JSON da solicitação

- **name** (*string*) – nome do componente
- **slug** (*string*) – URL amigável do componente
- **repo** (*string*) – URL do repositório VCS

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{
    "name": "new name"
}
```

Exemplo de resposta JSON:

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

{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "new name",
    "slug": "weblate",
    "project": {
        "name": "Hello",
        "slug": "hello",
        "source_language": {
```

(continues on next page)

(continuação da página anterior)

```
            "code": "en",
            "direction": "ltr",
            "population": 159034349015,
            "name": "English",
            "url": "http://example.com/api/languages/en/",
            "web_url": "http://example.com/languages/en/"
        },
        "url": "http://example.com/api/projects/hello/",
        "web": "https://weblate.org/",
        "web_url": "http://example.com/projects/hello/"
    },
    "repo": "file:///home/nijel/work/weblate-hello",
    "template": "",
    "new_base": "",
    "url": "http://example.com/api/components/hello/weblate/",
    "vcs": "git",
    "web_url": "http://example.com/projects/hello/weblate/"
}
```

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

Edita um componente por uma solicitação PUT.

Parâmetros

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

Objeto JSON da solicitação

- **branch** (*string*) – Ramo do repositório VCS
- **file_format** (*string*) – formato de ficheiro de traduções
- **filemask** (*string*) – máscara dos ficheiros de tradução no repositório
- **name** (*string*) – nome do componente
- **slug** (*string*) – URL amigável do componente
- **repo** (*string*) – URL do repositório VCS
- **template** (*string*) – ficheiro base para traduções monolíngues
- **new_base** (*string*) – ficheiro base para adicionar novas traduções
- **vcs** (*string*) – sistema de controle de versão

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

Novo na versão 3.9.

Apaga um componente.

Parâmetros

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

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

Retorna uma lista de alterações de componentes. Este é essencialmente um componente com escop *GET /api/changes/* a aceitar os mesmos parâmetros.

Parâmetros

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

Objeto JSON de resposta

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

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

Novo na versão 4.9.

Descarrega odas as traduções disponíveis associadas ao componente como um ficheiro a usar o formato solicitado.

Parâmetros

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

Parâmetros de consulta

- **format** (*string*) – O formato de ficheiro a ser usado; Se não for especificado, o padrão é `zip`; Formatos suportados: `zip`

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

Retorna uma lista de capturas de ecrã do componente.

Parâmetros

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

Objeto JSON de resposta

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

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

Retorna o estado do bloqueio do componente.

Parâmetros

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

Objeto JSON de resposta

- **locked** (*boolean*) – se o componente está bloqueado para atualizações

Exemplo de dados JSON:

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

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

Define o estado de bloqueio do componente.

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

Parâmetros

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

Objeto JSON da solicitação

- **lock** – Booleano se deve ser bloqueado ou não.

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{"lock": true}
```

Exemplo de resposta JSON:

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

{"locked":true}
```

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

Retorna informações sobre o estado do repositório VCS.

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

Parâmetros

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

Objeto JSON de resposta

- **needs_commit** (*boolean*) – se há alguma alteração pendente de commit
- **needs_merge** (*boolean*) – se há alguma alteração do upstream para mesclar
- **needs_push** (*boolean*) – se há alguma alteração local para fazer push
- **remote_commit** (*string*) – Informações do commit remoto
- **status** (*string*) – Estado do repositório VCS como relatado pelo VCS
- **merge_failure** – Texto a descrever uma falha de mesclagem ou nulo se houver nenhuma

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

Executa a operação fornecida num repositório VCS.

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

Parâmetros

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

Objeto JSON da solicitação

- **operation** (*string*) – Operação para executar: um de `push`, `pull`, `commit`, `reset`, `cleanup`

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{"operation":"pull"}
```

Exemplo de resposta JSON:

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

{"result":true}
```

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

Descarrega o ficheiro base para traduções monolíngues.

Parâmetros

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

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

Descarrega o ficheiro de modelo para novas traduções.

Parâmetros

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

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

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

Retorna uma lista de objetos de tradução no componente fornecido.

Parâmetros

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de tradução; veja *GET /api/translations/(string:project)/(string:component)/(string:language)/*

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

Cria uma nova tradução no componente fornecido.

Parâmetros

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

Objeto JSON da solicitação

- **language_code** (*string*) – código de idioma de tradução; veja *GET /api/languages/(string:language)/*

Objeto JSON de resposta

- **result** (*object*) – novo objecto de tradução criado

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{"language_code": "cs"}
```

Exemplo de resposta JSON:

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

{
    "failing_checks": 0,
```

(continues on next page)

(continuação da página anterior)


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


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

Retorna estatísticas paginadas para todas as traduções dentro do componente.

Novo na versão 2.7.

Parâmetros

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de estatísticas de tradução; veja *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/*

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

Retorna projetos vinculados a um componente.

Novo na versão 4.5.

Parâmetros

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

Objeto JSON de resposta

- **projects** (*array*) – projetos associados; veja *GET /api/projects/(string:project)/*

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

Associa projeto a um componente.

Novo na versão 4.5.

Parâmetros

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

Parâmetros de formulário

- **string project_slug** – URL amigável do projeto

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

Remove associação de um projeto com um componente.

Novo na versão 4.5.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **project_slug** (*string*) – Slug do projeto para remover

1.12.9 Traduções

GET /api/translations/

Devolve uma lista de traduções.

Veja também:

Atributos de objeto de tradução estão documentados em *GET /api/translations/(string:project)/(string:component)/(string:language)/*.

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

Devolve informação sobre uma tradução.

Parâmetros

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

Objeto JSON de resposta

- **component** (*object*) – objeto do componente; veja *GET /api/components/(string:project)/(string:component)/*
- **failing_checks** (*int*) – quantidade de cadeias com verificações falhadas
- **failing_checks_percent** (*float*) – Cadeias traduzidas com quaisquer verificações falhadas
- **failing_checks_words** (*int*) – quantidade de verificações falhadas
- **filename** (*string*) – Nome do ficheiro da tradução
- **fuzzy** (*int*) – quantidade de cadeias aproximadas (marcados para edição)

- **fuzzy_percent** (*float*) – percentagem de cadeias aproximadas (marcados para edição)
- **fuzzy_words** (*int*) – quantidade de palavras em cadeias aproximadas (marcados para edição)
- **have_comment** (*int*) – quantidade de cadeias com comentário
- **have_suggestion** (*int*) – quantidade de cadeias com sugestão
- **is_template** (*boolean*) – se a tradução tem uma base monolingue
- **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 úlitmo autor
- **last_change** (*timestamp*) – data e hora da última alteração
- **revision** (*string*) – hash de revisão para o ficheiro
- **share_url** (*string*) – URL para compartilhar a levar à página de engajamento
- **total** (*int*) – quantidade total de cadeias
- **total_words** (*int*) – número total de palavras
- **translate_url** (*string*) – URL para tradução
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **translated_words** (*int*) – número de palavras traduzidas
- **repository_url** (*string*) – URL para estado do repositório; veja *GET /api/translations/(string:project)/(string:component)/(string:language)/repository/*
- **file_url** (*string*) – URL para objeto ficheiro; veja *GET /api/translations/(string:project)/(string:component)/(string:language)/file/*
- **changes_list_url** (*string*) – URL para lista de alterações; veja *GET /api/translations/(string:project)/(string:component)/(string:language)/changes/*
- **units_list_url** (*string*) – URL para lista de cadeias; veja *GET /api/translations/(string:project)/(string:component)/(string:language)/units/*

Exemplo de dados JSON:

```
{
    "component": {
        "branch": "main",
        "file_format": "po",
        "filemask": "po/*.po",
        "git_export": "",
        "license": "",
        "license_url": "",
        "name": "Weblate",
        "new_base": "",
        "project": {
            "name": "Hello",
            "slug": "hello",
```

(continues on next page)

(continuação da página anterior)


```
            "source_language": {
                "code": "en",
                "direction": "ltr",
                "population": 159034349015,
                "name": "English",
                "url": "http://example.com/api/languages/en/",
                "web_url": "http://example.com/languages/en/"
            },
            "url": "http://example.com/api/projects/hello/",
            "web": "https://weblate.org/",
            "web_url": "http://example.com/projects/hello/"
        },
        "repo": "file:///home/nijel/work/weblate-hello",
        "slug": "weblate",
        "template": "",
        "url": "http://example.com/api/components/hello/weblate/",
        "vcs": "git",
        "web_url": "http://example.com/projects/hello/weblate/"
    },
    "failing_checks": 3,
    "failing_checks_percent": 75.0,
    "failing_checks_words": 11,
    "filename": "po/cs.po",
    "fuzzy": 0,
    "fuzzy_percent": 0.0,
    "fuzzy_words": 0,
    "have_comment": 0,
    "have_suggestion": 0,
    "is_template": false,
    "language": {
        "code": "cs",
        "direction": "ltr",
        "population": 1303174280
        "name": "Czech",
        "url": "http://example.com/api/languages/cs/",
        "web_url": "http://example.com/languages/cs/"
    },
    "language_code": "cs",
    "last_author": "Weblate Admin",
    "last_change": "2016-03-07T10:20:05.499",
    "revision": "7ddfafe6daaf57fc8654cc852ea6be212b015792",
    "share_url": "http://example.com/engage/hello/cs/",
    "total": 4,
    "total_words": 15,
    "translate_url": "http://example.com/translate/hello/weblate/cs/",
    "translated": 4,
    "translated_percent": 100.0,
    "translated_words": 15,
    "url": "http://example.com/api/translations/hello/weblate/cs/",
    "web_url": "http://example.com/projects/hello/weblate/cs/"
}
```


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

Novo na versão 3.9.

Apaga uma tradução.

Parâmetros

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

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

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

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

Parâmetros

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

Objeto JSON de resposta

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

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

Retorna uma lista de unidades de tradução.

Parâmetros

- **project** (*string*) – URL semântico do projeto
- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução
- **q** (*string*) – Texto de consulta de pesquisa; veja *Pesquisando* (opcional)

Objeto JSON de resposta

- **results** (*array*) – texto de objetos de componente; veja `GET /api/units/(int:id)/`

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

Add new unit.

Parâmetros

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

Objeto JSON da solicitação

- **key** (*string*) – Name of translation unit (used as key or context)
- **value** (*array*) – Textos fonte (use um único texto se não estiver criando plural)

Objeto JSON de resposta

- **unit** (*object*) – newly created unit; see `GET /api/units/(int:id)/`

Veja também:

Gerir cadeias, adding-new-strings

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

Aciona a tradução automática.

Parâmetros

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

- **component** (*string*) – URL semântico do componente
- **language** (*string*) – Código de idioma de tradução

Objeto JSON da solicitação

- **mode** (*string*) – Modo de tradução automático
- **filter_type** (*string*) – Tipo de filtro de tradução automática
- **auto_source** (*string*) – Fonte da tradução automática – `mt` ou `others`
- **component** (*string*) – Ativar a contribuição para a memória de tradução compartilhada para que o projeto tenha acesso a componentes adicionais.
- **engines** (*array*) – Motores de tradução automática
- **threshold** (*string*) – Limite de pontuação

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

Descarrega o ficheiro de tradução atual como está armazenado no VCS (sem o parâmetro `format`) ou convertido para outro formato (veja *Descarregar traduções*).

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

Parâmetros de consulta

- **format** – Formato de ficheiro a usar; se não for especificado nenhuma conversão de formato acontecerá; formatos de ficheiro suportados: `po`, `mo`, `xliff`, `xliff11`, `tbx`, `csv`, `xlsx`, `json`, `aresource`, `strings`
- **q** (*string*) – Filtra os textos descarregados, veja search.

Parâmetros

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

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

Envia um novo ficheiro com traduções.

Parâmetros

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

Parâmetros de formulário

- **string conflict** – Como lidar com conflitos (`ignore`, `replace-translated` ou `replace-approved`)
- **file file** – Ficheiro enviado
- **string email** – E-mail do autor
- **string author** – Nome do autor
- **string method** – Método de envio (`translate`, `approve`, `suggest`, `fuzzy`, `replace`, `source`, `add`), veja *Métodos de importação*

- **string fuzzy** – Processamento de cadeia aproximadas (marcados para edição) (*vazio*, `process`, `approve`)

Exemplo de CURL:

```
curl -X POST \
    -F file=@strings.xml \
    -H "Authorization: Token TOKEN" \
    http://example.com/api/translations/hello/android/cs/file/
```

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

Retorna informações sobre o estado do repositório VCS.

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

Parâmetros

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

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

Realiza a operação dada no repositório VCS.

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

Parâmetros

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

Objeto JSON da solicitação

- **operation** (*string*) – Operação para executar: um de `push`, `pull`, `commit`, `reset`, `cleanup`

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

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

Devolve estatísticas detalhadas da tradução.

Novo na versão 2.7.

Parâmetros

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

Objeto JSON de resposta

- **code** (*string*) – código de idioma
- **failing** (*int*) – número de verificações falhadas
- **failing_percent** (*float*) – percentagem de verificações com falha
- **fuzzy** (*int*) – quantidade de cadeias aproximadas (marcados para edição)

- **fuzzy_percent** (*float*) – percentagem de cadeias aproximadas (marcados para edição)
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas
- **last_author** (*string*) – nome do úlitmo autor
- **last_change** (*timestamp*) – data da última alteração
- **name** (*string*) – nome de idioma
- **total** (*int*) – quantidade total de cadeias
- **translated** (*int*) – quantidade de cadeias traduzidas
- **translated_percent** (*float*) – percentagem de cadeia traduzidas
- **url** (*string*) – URL para acesso à tradução (URL de engajamento)
- **url_translate** (*string*) – URL para acesso à tradução (URL real da tradução)

1.12.10 Memória

Novo na versão 4.14.

GET /api/memory/

Returns a list of memory results.

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

Deletes a memory object

Parâmetros

- **memory_object_id** – Memory Object ID

1.12.11 Unidades

Uma *unidade* é uma única parte de uma tradução que forma o par de um texto fonte com uma cadeia traduzida correspondente e também contém alguns metadados relacionados. O termo é derivado do Translate Toolkit e do XLIFF.

Novo na versão 2.10.

GET /api/units/

Retorna uma lista de unidades de tradução.

Parâmetros

- **q** (*string*) – Texto de consulta de pesquisa; veja *Pesquisando* (opcional)

Veja também:

Os atributos do objeto da unidade estão documentados em `GET /api/units/(int:id)/`.

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

Alterado na versão 4.3: O `target` e `source` agora são vetores para lidar apropriadamente com cadeias plurais.

Devolve informação sobre a unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON de resposta

- **translation** (*string*) – URL de um objeto de tradução relacionado
- **source** (*array*) – cadeia fonte
- **previous_source** (*string*) – cadeia fonte anterior usado para correspondência aproximada
- **target** (*array*) – cadeia alvo
- **id_hash** (*string*) – identificador único da unidade
- **content_hash** (*string*) – identificador único da cadeia fonte
- **location** (*string*) – local da unidade no código-fonte
- **context** (*string*) – contexto da unidade de tradução
- **note** (*string*) – nota da unidade de tradução
- **flags** (*string*) – marcadores da unidade de tradução
- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved, 100 - read only
- **fuzzy** (*boolean*) – se a unidade está confusa ou marcada para revisão
- **translated** (*boolean*) – Percentagem traduzido
- **approved** (*boolean*) – Tradução aprovada
- **position** (*int*) – posição da unidade no ficheiro de tradução
- **has_suggestion** (*boolean*) – Cadeia tem sugestão
- **has_comment** (*boolean*) – Cadeia tem comentário
- **has_failing_check** (*boolean*) – Cadeia tem verificação falhada
- **num_words** (*int*) – número de palavras de origem
- **priority** (*int*) – prioridade de tradução. 100 é o padrão
- **id** (*int*) – identificador da unidade
- **explanation** (*string*) – Explicação de cadeia, disponível em unidades fonte, veja *Informações adicionais sobre cadeias fonte*
- **extra_flags** (*string*) – Sinalizadores de cadeias adicionais, disponíveis nas unidades de fonte, veja *Personalizando comportamento a usar marcadores*
- **web_url** (*string*) – URL onde a unidade pode ser editada
- **source_unit** (*string*) – Ligação da unidade fonte; veja `GET /api/units/(int:id)/`
- **pending** (*boolean*) – whether the unit is pending for write
- **timestamp** (*timestamp*) – string age

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

Novo na versão 4.3.

Realiza uma atualização parcial na unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON da solicitação

- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see *Revisores dedicados*)
- **target** (*array*) – cadeia alvo

- **explanation** (*string*) – Explicação de cadeia, disponível em unidades fonte, veja *Informações adicionais sobre cadeias fonte*
- **extra_flags** (*string*) – Sinalizadores de cadeias adicionais, disponíveis nas unidades de fonte, veja *Personalizando comportamento a usar marcadores*

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

Novo na versão 4.3.

Realiza a atualização completa da unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON da solicitação

- **state** (*int*) – unit state, 0 - untranslated, 10 - needs editing, 20 - translated, 30 - approved (need review workflow enabled, see *Revisores dedicados*)
- **target** (*array*) – cadeia alvo
- **explanation** (*string*) – Explicação de cadeia, disponível em unidades fonte, veja *Informações adicionais sobre cadeias fonte*
- **extra_flags** (*string*) – Sinalizadores de cadeias adicionais, disponíveis nas unidades de fonte, veja *Personalizando comportamento a usar marcadores*

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

Novo na versão 4.3.

Apaga a unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

1.12.12 Alterações

Novo na versão 2.10.

GET /api/changes/

Alterado na versão 4.1: A filtragem de alterações foi introduzida na versão 4.1.

Retorna uma lista de alterações de tradução.

Veja também:

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

Parâmetros de consulta

- **user** (*string*) – Nome de utilizador do utilizador para filtros
- **action** (*int*) – Ação para filtrar, pode ser usada várias vezes
- **timestamp_after** (*timestamp*) – Data e hora no formato ISO 8601 para listar as alterações após
- **timestamp_before** (*timestamp*) – Data e hora no formato ISO 8601 para listar as alterações antes

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

Devolve informação sobre mudanças na tradução.

Parâmetros

- **id** (*int*) – ID da alteração

Objeto JSON de resposta

- **unit** (*string*) – URL de um objeto de unidade relacionado
- **translation** (*string*) – URL de um objeto de tradução relacionado
- **component** (*string*) – URL de um objeto de componente relacionado
- **user** (*string*) – URL de um objeto de utilizador relacionado
- **author** (*string*) – URL de um objeto de autor relacionado
- **timestamp** (*timestamp*) – data e hora do evento
- **action** (*int*) – identificação numérica da ação
- **action_name** (*string*) – descrição textual da ação
- **target** (*string*) – texto ou detalhe alterado de evento
- **id** (*int*) – identificador da alteração

1.12.13 Capturas de ecrã

Novo na versão 2.14.

GET /api/screenshots/

Retorna uma lista de informações de cadeias de capturas de ecrã.

Veja também:

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

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

Retorna informações sobre dados de captura de ecrã.

Parâmetros

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

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja *GET /api/units/(int:id)/*

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

Descarrega a imagem de captura de ecrã.

Parâmetros

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

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

Substitui a imagem de captura de ecrã.

Parâmetros

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

Parâmetros de formulário

- **file image** – Ficheiro enviado

Exemplo de CURL:

```
curl -X POST \
    -F image=@image.png \
    -H "Authorization: Token TOKEN" \
    http://example.com/api/screenshots/1/file/
```

POST /api/screenshots/(int: *id*)/units/

Associa cadeias fonte com uma captura de ecrã.

Parâmetros

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

Parâmetros de formulário

- **string unit_id** – ID da unidade

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **translation** (*string*) – URL de um objeto de tradução relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja *GET /api/units/(int:id)/*

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

Remove a associação da cadeia fonte com captura de ecrã.

Parâmetros

- **id** (*int*) – ID da captura de ecrã
- **unit_id** – ID da unidade da cadeia fonte

POST /api/screenshots/

Cria uma nova captura de ecrã.

Parâmetros de formulário

- **file image** – Ficheiro enviado
- **string name** – Nome da captura do ecrã
- **string project_slug** – URL amigável do projeto
- **string component_slug** – URL amigável do componente
- **string language_code** – Código do idioma

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja *GET /api/units/(int:id)/*

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

Edita informações parciais sobre captura de ecrã.

Parâmetros

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

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja *GET /api/units/(int:id)/*

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

Edita informações completas sobre captura de ecrã.

Parâmetros

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

Objeto JSON de resposta

- **name** (*string*) – nome de um screenshot
- **component** (*string*) – URL de um objeto de componente relacionado
- **file_url** (*string*) – URL para descarregar um ficheiro; veja *GET /api/screenshots/(int:id)/file/*
- **units** (*array*) – ligação para associar informações sobre cadeia fonte; veja *GET /api/units/(int:id)/*

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

Apagar captura de ecrã.

Parâmetros

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

1.12.14 Extensões

Novo na versão 4.4.1.

GET /api/addons/

Retorna uma lista de extensões.

Veja também:

Os atributos do objeto de extensão estão documentados em *GET /api/addons/(int:id)/*.

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

Retorna informações sobre dados da extensão.

Parâmetros

- **id** (*int*) – ID da extensão

Objeto JSON de resposta

- **name** (*string*) – nome de uma extensão
- **component** (*string*) – URL de um objeto de componente relacionado
- **configuration** (*object*) – Configuração opcional da extensão

Veja também:

Extensões

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

Cria uma nova extensão.

Parâmetros

- **project_slug** (*string*) – URL amigável do projeto
- **component_slug** (*string*) – URL amigável do componente

Objeto JSON da solicitação

- **name** (*string*) – nome de uma extensão
- **configuration** (*object*) – Configuração opcional da extensão

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

Edita informações parciais sobre extensão.

Parâmetros

- **id** (*int*) – ID da extensão

Objeto JSON de resposta

- **configuration** (*object*) – Configuração opcional da extensão

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

Edita informações completas sobre extensão.

Parâmetros

- **id** (*int*) – ID da extensão

Objeto JSON de resposta

- **configuration** (*object*) – Configuração opcional da extensão

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

Exclui extensão.

Parâmetros

- **id** (*int*) – ID da extensão

1.12.15 Listas de componentes

Novo na versão 4.0.

GET /api/component-lists/

Retorna uma lista de listas de componentes.

Veja também:

Os atributos do objeto da lista de componentes estão documentados em *GET /api/component-lists/(str:slug)/*.

GET /api/component-lists/(str: *slug*)/

Retorna informações sobre a lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON de resposta

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar num painel

- **components** (*array*) – ligação para componentes associados; veja *GET /api/components/(string:project)/(string:component)/*
- **auto_assign** (*array*) – regras de atribuição automáticas

PUT /api/component-lists/(str: *slug*)/

Altera os parâmetros da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON da solicitação

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar num painel

PATCH /api/component-lists/(str: *slug*)/

Altera os parâmetros da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON da solicitação

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar num painel

DELETE /api/component-lists/(str: *slug*)/

Apaga a lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

POST /api/component-lists/(str: *slug*)/components/

Associa um componente a uma lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Parâmetros de formulário

- **string component_id** – ID do componente

DELETE /api/component-lists/(str: *slug*)/components/(str: *component_slug*)

Desassocia um componente da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes
- **component_slug** (*string*) – URL amigável do componente

1.12.16 Glossário

Alterado na versão 4.5: Glossários agora são armazenados como componentes, traduções e cadeias comuns. Use a respetiva API.

1.12.17 Tarefas

Novo na versão 4.4.

GET /api/tasks/

A listagem de tarefas não está disponível no momento.

GET /api/tasks/(str: *uuid*)/

Retorna informações sobre uma tarefa

Parâmetros

- **uuid** (*string*) – UUID da tarefa

Objeto JSON de resposta

- **completed** (*boolean*) – Se a tarefa foi concluída
- **progress** (*int*) – Progresso da tarefa em percentagem
- **result** (*object*) – Resultado da tarefa ou detalhes do progresso
- **log** (*string*) – Registos da tarefa

1.12.18 Métricas

GET /api/metrics/

Retorna as métricas do servidor.

Objeto JSON de resposta

- **units** (*int*) – Quantidade de unidades
- **units_translated** (*int*) – Quantidade de unidades traduzidas
- **users** (*int*) – Quantidade de utilizadores
- **changes** (*int*) – Quantidade de alterações
- **projects** (*int*) – Quantidade de projetos
- **components** (*int*) – Quantidade de componentes
- **translations** (*int*) – Quantidade de traduções
- **languages** (*int*) – Quantidade de idiomas usados
- **checks** (*int*) – Quantidade de verificações de qualidade acionadas
- **configuration_errors** (*int*) – Quantidade de erros de configuração
- **suggestions** (*int*) – Quantidade de sugestões pendentes
- **celery_queues** (*object*) – Comprimento das filas de Celery, veja *Tarefas de fundo a usar o Celery*
- **name** (*string*) – Nome do servidor configurado

1.12.19 Hooks de notificação

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

Pode usar terminais de repositório para projetos, componentes e traduções para atualizar repositórios individuais; veja *POST /api/projects/(string:project)/repository/* para a documentação.

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

Obsoleto desde a versão 2.6: Use *POST /api/components/(string:project)/(string:component)/repository/* em vez disso, que funciona corretamente com autenticação para projetos limitados por ACL.

Aciona a atualização de um componente (a buscar do VCS e a procurar por alterações de tradução).

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

Obsoleto desde a versão 2.6: Use *POST /api/projects/(string:project)/repository/*, que funciona corretamente com autenticação para projetos limitados por ACL.

Aciona a atualização de todos os componentes num projeto (a descarregar do VCS e a procurar por alterações de tradução).

POST /hooks/github/

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

Nota: O GitHub inclui suporte direto para notificar o Weblate: ative o gancho do serviço Weblate nas configurações do repositório e defina a URL para a URL da sua instalação do Weblate.

Veja também:

Receber alterações do GitHub automaticamente
Para instruções sobre como configurar integração com GitHub

https://docs.github.com/en/get-started/customizing-your-github-workflow/exploring-integrations/about-webhooks
Informações genéricas sobre Webhooks do GitHub

ENABLE_HOOKS
Para ativar ganchos para todo o Weblate

POST /hooks/gitlab/

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

Veja também:

Receber alterações do GitLab automaticamente
Para instrução sobre configurar integração com GitLab

https://docs.gitlab.com/ee/user/project/integrations/webhooks.html
Informações genéricas sobre Webhooks do GitLab

ENABLE_HOOKS
Para ativar ganchos para todo o Weblate

POST /hooks/bitbucket/

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

Veja também:

Receber alterações do Bitbucket automaticamente
Para instruções sobre como configurar integração com Bitbucket

https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/
Informações genéricas sobre Webhooks do Bitbucket

ENABLE_HOOKS
Para ativar ganchos para todo o Weblate

POST /hooks/pagure/

Novo na versão 3.3.

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

Veja também:

Receber alterações do Pagure automaticamente
Para instruções sobre como configurar integração com Pagure

https://docs.pagure.org/pagure/usage/using_webhooks.html
Informações genéricas sobre Webhooks do Pagure

ENABLE_HOOKS
Para ativar ganchos para todo o Weblate

POST /hooks/azure/

Novo na versão 3.8.

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

Nota: Certifique-se de que *Resource details to send* esteja definido como *All*, caso contrário o Weblate não poderá corresponder ao seu repositório do Azure.

Veja também:

Receber alterações dos Azure Repos automaticamente
Para instruções sobre como configurar integração com Azure

https://docs.microsoft.com/pt-br/azure/devops/service-hooks/services/webhooks?view=azure-devops
Generic information about Azure DevOps Web Hooks

ENABLE_HOOKS
Para ativar ganchos para todo o Weblate

POST /hooks/gitea/

Novo na versão 3.9.

Gancho especial para lidar com notificações de Webhook do Gitea e atualizar automaticamente os componentes correspondentes.

Veja também:

Receber alterações dos Gitea Repos automaticamente
Para instruções sobre como configurar integração com Gitea

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

ENABLE_HOOKS
Para ativar ganchos para todo o Weblate

POST /hooks/gitee/

Novo na versão 3.9.

Gancho especial para lidar com notificações de Webhook do Gitee e atualizar automaticamente os componentes correspondentes.

Veja também:

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

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

ENABLE_HOOKS
Para ativar ganchos para todo o Weblate

1.12.20 Exportações

O Weblate fornece várias exportações para permitir que processe ainda mais os dados.

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

Parâmetros de consulta

- **format** (*string*) – Formato de saída: `json` ou `csv`

Obsoleto desde a versão 2.6: Use *GET /api/components/(string:project)/(string:component)/statistics/* e *GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/* em vez disso; ele também permite acesso a projetos controlados por ACL.

Recupera estatísticas para determinado componente em determinado formato.

Exemplo de solicitação:

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

Exemplo de resposta:

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

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

(continues on next page)

(continuação da página anterior)


```
        "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/"
    }
]
```


1.12.21 Feeds RSS

As alterações nas traduções são exportadas em feeds RSS.

GET /exports/rss/(string: *project*)/(string: *component*)/(string: *language*)/

Recupera o feed RSS com as alterações recentes de uma tradução.

GET /exports/rss/(string: *project*)/(string: *component*)/

Recupera feed RSS com alterações recentes para um componente.

GET /exports/rss/(string: *project*)/

Recupera feed RSS com alterações recentes para um projeto.

GET /exports/rss/language/(string: *language*)/

Recupera feed RSS com alterações recentes para um idioma.

GET /exports/rss/

Recupera feed RSS com alterações recentes para a instância Weblate.

Veja também:

RSS on Wikipedia

1.13 Cliente Weblate

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

1.13.1 Instalação

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

```
pip install wlc
```

1.13.2 Uso do Docker

O Weblate Client também está disponível como uma imagem Docker.

A imagem está publicada no Docker Hub: https://hub.docker.com/r/weblate/wlc

Instalar:

```
docker pull weblate/wlc
```

O contentor Docker usa as configurações padrão do Weblate e se conecta à API implementada no localhost. A URL da API e API_KEY podem ser configuradas através dos argumentos aceitos pelo Weblate.

O comando para iniciar o contentor usa a seguinte sintaxe:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Exemplo:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

Quereria passar o seu *Ficheiros de configuração* para o contentor Docker, a abordagem mais fácil é adicionar o seu diretório atual como o volume `/home/weblate`:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Primeiros Passos

A configuração do wlc é armazenada em ``~/.config/weblate``(veja *Ficheiros de configuração* para outros locais), por favor, crie-a para corresponder ao seu ambiente:

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

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

Depois pode invocar comandos no servidor predefinido:

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

Veja também:

Ficheiros de configuração

1.13.4 Sinopse

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

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

1.13.5 Descrição

O cliente Weblate é uma biblioteca Python e utilitário de linha de comando para gerir o Weblate remotamente a usar a *API REST do Weblate*. O utilitário de linha de comando pode ser invocado como **wlc** e está embutido em `wlc`.

Argumentos

O programa aceita os seguintes argumentos que definem o formato de saída ou qual a instância do Weblate a utilizar. Estes devem ser introduzidos antes de qualquer comando.

--format `{csv,json,text,html}`

Especifica o formato de saída.

--url `URL`

Especifica a URL da API. Substitui qualquer valor encontrado no ficheiro de configuração, consulte *Ficheiros de configuração*. A URL deve terminar com `/api/`, por exemplo, `https://hosted.weblate.org/api/`.

--key `KEY`

Especifica a chave do utilizador de API a ser usada. Substitui qualquer valor encontrado no ficheiro de configuração, consulte *Ficheiros de configuração*. Pode encontrar a sua chave no seu perfil no Weblate.

--config `PATH`

Substitui o caminho do ficheiro de configuração, consulte *Ficheiros de configuração*.

--config-section `SECTION`

Substitui a secção de ficheiros de configuração em uso, consulte *Ficheiros de configuração*.

Comandos

Os comandos seguintes estão disponíveis:

version

Imprime a versão atual.

list-languages

Lista os idiomas usados no Weblate.

list-projects

Lista os projetos no Weblate.

list-components

Lista os componentes no Weblate.

list-translations

Lista as traduções no Weblate.

show

Mostra o objeto do Weblate (tradução, componente ou projeto).

ls

Lista o objeto do Weblate (tradução, componente ou projeto).

commit

Faz um commit das alterações feitas num objeto Weblate (tradução, componente ou projeto).

pull

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

push

Faz um push das alterações do objeto Weblate ao repositório remoto (tradução, componente ou projeto).

reset

Novo na versão 0.7: Suportado desde o wlc 0.7.

Redefine as alterações no objeto Weblate para corresponder ao repositório remoto (tradução, componente ou projeto).

cleanup

Novo na versão 0.9: Suportado desde o wlc 0.9.

Remove todas as alterações não rastreadas num objeto Weblate para corresponder ao repositório remoto (tradução, componente ou projeto).

repo

Exibe o estado do repositório para um determinado objeto do Weblate (tradução, componente ou projeto).

stats

Exibe estatísticas detalhadas para um determinado objeto Weblate (tradução, componente ou projeto).

lock-status

Novo na versão 0.5: Suportado desde o wlc 0.5.

Exibe o estado do bloqueio.

lock

Novo na versão 0.5: Suportado desde o wlc 0.5.

Bloqueia o componente de tradução posterior no Weblate.

unlock

Novo na versão 0.5: Suportado desde o wlc 0.5.

Desbloqueia a tradução do componente Weblate.

changes

Novo na versão 0.7: Suportado desde o wlc 0.7 e o Weblate 2.10.

Exibe alterações para um determinado objeto.

download

Novo na versão 0.7: Suportado desde o wlc 0.7.

Descarrega um ficheiro de tradução.

--convert

Converte o formato do ficheiro, se nenhuma conversão não especificada for feita no servidor e o ficheiro for descarregado como está no repositório.

--output

Especifica o ficheiro para gravar a saída e se não for especificado é impresso na stdout (saída predefinida).

upload

Novo na versão 0.9: Suportado desde o wlc 0.9.

Descarrega um ficheiro de tradução.

--overwrite

Substitua as traduções existentes ao enviar.

--input

Ficheiro a partir do qual o conteúdo é lido, se não for especificado é lido de stdin (entrada predefinida).

--method

Método de envio para usar, veja *Métodos de importação*.

--fuzzy

Processamento de cadeia aproximadas (marcados para edição) (*vazio*, `process`, `approve`)

--author-name

Nome do autor, para substituir o utilizador atualmente autenticado

--author-email

E-mail do autor, para substituir o utilizador atualmente autenticado

---

Dica: Pode obter informações mais detalhadas sobre como invocar comandos individuais a passar `--help`, por exemplo: `wlc ls --help`.

1.13.6 Ficheiros de configuração

.weblate, .weblate.ini, weblate.ini

Alterado na versão 1.6: Os ficheiros com extensão *.ini* também são aceitos.

Por ficheiro de configuração de projeto

C:\Users\NAME\AppData\weblate.ini

Novo na versão 1.6.

Ficheiro de configuração do utilizador no Windows.

~/.config/weblate

Ficheiro de configuração do utilizador

/etc/xdg/weblate

Ficheiro de configuração para todo o sistema

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

As configurações seguintes podem ser configuradas na secção `[weblate]` (pode personalizar-lo por *`--config-section`*):

key

Chave de API para acessar o Weblate.

url

URL de API do servidor, a predefinição é `http://127.0.0.1:8000/api/`.

translation

Caminho à tradução predefinida - componente ou projeto.

O ficheiro de configuração é um ficheiro INI, por exemplo:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Além disso, as chaves de API podem ser armazenadas na secção `[keys]`:

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

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

1.13.7 Exemplos

Imprimir a versão atual do programa:

```
$ wlc version
version: 0.1
```

Listar todos os projetos:

```
$ wlc list-projects
name: Hello
slug: hello
url: http://example.com/api/projects/hello/
web: https://weblate.org/
web_url: http://example.com/projects/hello/
```

Enviar ficheiro de tradução:

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

Também pode designar em qual projeto o wlc deve trabalhar:

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

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

Com esta configuração é fácil fazer um commit de alterações pendentes no projeto atual:

```
$ wlc commit
```

1.14 API Python do Weblate

1.14.1 Instalação

A API Python é distribuída separadamente, precisa instalar o *Cliente Weblate* (wlc) para tê-la.

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception wlc.**WeblateException**

Classe base para todas as exceções.

Weblate

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

Parâmetros

- **key** (*str*) – Chave do utilizador
- **url** (*str*) – URL da API do servidor; se não especificado, o padrão é usado
- **config** (wlc.config.WeblateConfig) – Objeto de configuração, substitui quaisquer outros parâmetros.

Acessa a classe para a API, define a chave da API e, opcionalmente, a URL da API.

get(*path*)

Parâmetros
path (*str*) – Caminho da solicitação

Tipo de retorno
object

Executa uma única chamada de API GET.

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

Parâmetros
path (*str*) – Caminho da solicitação

Tipo de retorno
object

Executa uma única chamada de API GET.

1.14.3 wlc.config

WeblateConfig

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

Parâmetros
section (*str*) – Seção de configuração para uso

Analisador de ficheiro de configuração a seguir a especificação XDG.

load(*path=None*)

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

Carrega a configuração de um ficheiro. Se nenhuma for especificada, carrega do ficheiro de configuração *wlc* (`~/.config/wlc`) posto no seu caminho de configuração XDG (`/etc/xdg/wlc`).

1.14.4 wlc.main

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

Parâmetros

- **settings** (*list*) – Configurações para substituir como lista de tuplas
- **stdout** (*object*) – objeto de ficheiro stdout para saída de impressão, usa `sys.stdout` como padrão
- **args** (*list*) – Argumentos de linha de comando para processar, usa `sys.args` como padrão

Ponto de entrada principal para interface de linha de comando.

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

Decorador para registar a classe `Command` no analisador principal usado por `main()`.

Command

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

Classe principal para invocação de comandos.

CAPÍTULO 2

Documentação de administrador

2.1 Instruções de configuração

2.1.1 Instalar o Weblate

Instalando a usar Docker

Com a implantação do Weblate dockerizada, pode pôr a sua instância Weblate pessoal em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão.

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

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

Instalação

Os exemplos a seguir presumem que tem um ambiente Docker funcional, com `docker-compose` instalado. Verifique a documentação do Docker para obter instruções.

1. Clone o repositório weblate-docker:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Crie um ficheiro `docker-compose.override.yml` com as suas configurações. Veja *Variáveis de ambiente do Docker* para uma lista completa das variáveis de ambiente.

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SERVER_EMAIL: weblate@example.com
      WEBLATE_DEFAULT_FROM_EMAIL: weblate@example.com
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for the admin user
      WEBLATE_ADMIN_EMAIL: weblate.admin@example.com
```

Nota: Se *`WEBLATE_ADMIN_PASSWORD`* não estiver definida, o utilizador admin é criado com uma palavra-passe aleatória mostrada na primeira inicialização.

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

3. Inicie os contentores do Weblate:

```
docker-compose up
```

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

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

Alterado na versão 3.7.1-6: Em julho de 2019 (a começar com a tag 3.7.1-6), os contentores não estão a ser executados como um utilizador root. Isso mudou a porta exposta de 80 para 8080.

Veja também:

Invocando comandos de gestão

Escolhendo a tag do hub Docker

Pode usar as seguintes tags no hub do Docker, veja https://hub.docker.com/r/weblate/weblate/tags/ para uma lista completa das tags disponíveis.

Nome da tag	Descrição	Caso de uso
`latest`	Versão estável do Weblate, corresponde à última versão marcada	Atualizações contínuas num ambiente de produção
`<VERSION>-<PATCH>`	Versão estável do Weblate	Implantação bem definida num ambiente de produção
`edge`	Lançamento estável do Weblate com alterações de desenvolvimento no contentor Docker (por exemplo, dependências atualizadas)	Atualizações contínuas num ambiente de teste
`edge-<DATE>-<CHAN`	Lançamento estável do Weblate com alterações de desenvolvimento no contentor Docker (por exemplo, dependências atualizadas)	Implantação bem definida num ambiente de teste
`bleeding`	Versão de desenvolvimento do Weblate do Git	Atualizações contínuas para testar os próximos recursos do Weblate
`bleeding-<DATE>-<CHA`	Versão de desenvolvimento do Weblate do Git	Implantação bem definida para testar os próximos recursos da Weblate

Cada imagem é testada pelo nosso CI antes de ser publicada, então até mesmo a versão *bleeding* deve ser bastante segura de usar.

Contentor Docker com suporte a HTTPS

Por favor, veja *Instalação* para instruções genéricas de implantação, esta secção apenas menciona diferenças em comparação a ela.

Usando os seus próprios certificados SSL

Novo na versão 3.8-3.

No caso de ter o seu próprio certificado SSL que deseja usar, basta pôr os ficheiros no volume de dados Weblate (veja *Volumes de contentor Docker*):

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

Ambos os ficheiros devem pertencer ao mesmo utilizador que inicia o contentor do docker e ter a máscara de ficheiro definida como `600` (legível e gravável apenas pelo utilizador dono).

Além disso, o contentor Weblate agora aceitará conexões SSL na porta 4443. Ainda quererá incluir o encaminhamento de porta para HTTPS na substituição de composição do docker:

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

Se já hospeda outros sites no mesmo servidor, é provável que as portas `80` e `443` sejam usadas por um proxy reverso, como NGINX. Para passar a conexão HTTPS do NGINX para o contentor do docker, pode usar a seguinte configuração:

```
server {
    listen 443;
    listen [::]:443;

    server_name <SITE_URL>;
    ssl_certificate /etc/letsencrypt/live/<SITE>/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/<SITE>/privkey.pem;

    location / {
            proxy_set_header HOST $host;
            proxy_set_header X-Forwarded-Proto https;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Host $server_name;
            proxy_pass https://127.0.0.1:<EXPOSED_DOCKER_PORT>;
    }
}
```

Substitua `<SITE_URL>`, `<SITE>` e `<EXPOSED_DOCKER_PORT>` por valores reais do seu ambiente.

Certificados SSL automáticos a usar Let's Encrypt

Caso queira usar certificados SSL Let's Encrypt gerados automaticamente na instalação pública, precisa adicionar um proxy HTTPS reverso num contentor Docker adicional, https-portal será usado para isso. Isso é usado no ficheiro `docker-compose-https.yml`. Em seguida, crie um ficheiro `docker-compose-https.override.yml` com as suas configurações:

```
version: '3'
services:
  weblate:
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for admin user
  https-portal:
    environment:
      DOMAINS: 'weblate.example.com -> http://weblate:8080'
```

Sempre que invocar **`docker-compose`**, precisa passar os dois ficheios a ele e então fazer:

```
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml␣
↪build
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml up
```

Atualizando o contentor Docker

Normalmente, é uma boa ideia atualizar apenas o contentor Weblate e manter o contentor PostgreSQL na versão que possui, já que atualizar o PostgreSQL é muito doloroso e na maioria dos casos não traz muitos benefícios.

Alterado na versão 4.10-1: Since Weblate 4.10-1, the Docker container uses Django 4.0 what requires PostgreSQL 10 or newer, please upgrade it prior to upgrading Weblate. See *Atualizar da 4.9 para 4.10* and *Upgrading PostgreSQL container*.

Pode fazer isso a manter o docker-compose existente e apenas obter as imagens mais recentes e reiniciar:

```
# Fetch latest versions of the images
docker-compose pull
# Stop and destroy the containers
docker-compose down
# Spawn new containers in the background
docker-compose up -d
# Follow the logs during upgrade
docker-compose logs -f
```

O banco de dados do Weblate deve ser migrado automaticamente na primeira inicialização e não deve haver necessidade de ações manuais adicionais.

Nota: Upgrades across major versions are not supported by Weblate. For example, if you are on 3.x series and want to upgrade to 4.x, first upgrade to the latest 4.0.x-y image (at time of writing this it is the `4.0.4-5`), which will do the migration and then continue upgrading to newer versions.

You might also want to update the `docker-compose` repository, though it's not needed in most case. See *Upgrading PostgreSQL container* for upgrading the PostgreSQL server.

Upgrading PostgreSQL container

Os contentores PostgreSQL não oferecem suporte a atualização automática entre versões, precisa realizar a atualização manualmente. Os passos a seguir mostram uma das opções de atualização.

Veja também:

https://github.com/docker-library/postgres/issues/37

1. Stop Weblate container:

```
docker-compose stop weblate cache
```

2. Backup the database:

```
docker-compose exec database pg_dumpall --clean --username weblate > backup.sql
```

3. Stop the database container:

```
docker-compose stop database
```

4. Remove the PostgreSQL volume:

```
docker-compose rm -v database
docker volume remove weblate_postgres-data
```

5. Adjust `docker-compose.yml` to use new PostgreSQL version.
6. Start the database container:

```
docker-compose up -d database
```

7. Restaure o banco de dados do backup:

```
cat backup.sql | docker-compose exec -T database psql --username weblate --
↪dbname postgres
```

8. Start all remaining containers:

```
docker-compose up -d
```

Autenticação como administrador

Após a configuração do contentor, pode entrar como utilizador *admin* com a palavra-passe fornecida em *WEBLATE_ADMIN_PASSWORD*, ou uma palavra-passe aleatória gerada na primeira inicialização se não tiver sido definida.

Para redefinir a palavra-passe do *admin*, reinicie o contentor com *WEBLATE_ADMIN_PASSWORD* definido com a nova palavra-passe.

Veja também:

WEBLATE_ADMIN_PASSWORD, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

Quantidade de processos e consumo de memória

A quantidade de processos de trabalho para uWSGI e Celery é determinado automaticamente com base no quantidade de CPUs. Isso funciona bem para a maioria das máquinas virtuais em nuvem, pois normalmente têm poucas CPUs e boa quantidade de memória.

Caso tenha muitos núcleos de CPU e tenha problemas de memória insuficiente, tente reduzir a quantidade de workers:

```
environment:
  WEBLATE_WORKERS: 2
```

Também pode ajustar as categorias de workers individuais:

```
environment:
  WEB_WORKERS: 4
  CELERY_MAIN_OPTIONS: --concurrency 2
  CELERY_NOTIFY_OPTIONS: --concurrency 1
  CELERY_TRANSLATE_OPTIONS: --concurrency 1
```

Veja também:

WEBLATE_WORKERS, *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS*, *WEB_WORKERS*

Dimensionando horizontalmente

Novo na versão 4.6.

Pode executar vários contentores Weblate para dimensionar o serviço horizontalmente. O volume `/app/data` deve ser compartilhado por todos os contentores, é recomendado usar um sistema de ficheiros de cluster como o GlusterFS para isso. O volume `/app/cache` deve ser separado para cada contentor.

Cada contentor Weblate tem um papel definido a usar a variável de ambiente *WEBLATE_SERVICE*. Siga atentamente a documentação, pois alguns dos serviços devem ser executados apenas uma vez no cluster e a ordem dos serviços também é importante.

Pode encontrar configuração de exemplo no repositório `docker-compose` como docker-compose-split.yml.

Variáveis de ambiente do Docker

Muitas das *Configurações* do Weblate podem ser definidas no contentor Docker a usar variáveis de ambiente:

Configurações genéricas

WEBLATE_DEBUG

Configura o modo de depuração do Django a usar `DEBUG`.

Exemplo:

```
environment:
  WEBLATE_DEBUG: 1
```

Veja também:

Desativar o modo de depuração

WEBLATE_LOGLEVEL

Configura o detalhamento do log.

WEBLATE_LOGLEVEL_DATABASE

Configures the logging of the database queries verbosity.

WEBLATE_SITE_TITLE

Altera o título do site mostrado no cabeçalho de todas as páginas.

WEBLATE_SITE_DOMAIN

Configura o domínio do site. Este parâmetro é obrigatório.

Veja também:

Definir domínio correto do site, `SITE_DOMAIN`

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Configura o nome e o e-mail do administrador do site. É usado para `ADMINS` e para criar o utilizador *admin* (veja `WEBLATE_ADMIN_PASSWORD` para mais informações).

Exemplo:

```
environment:
  WEBLATE_ADMIN_NAME: Weblate admin
  WEBLATE_ADMIN_EMAIL: noreply@example.com
```

Veja também:

Autenticação como administrador, *Configurar administradores corretamente*, `ADMINS`

WEBLATE_ADMIN_PASSWORD

Define a palavra-passe para o utilizador *admin*.

- Se não for definido e o utilizador *admin* não existir, ele será criado com uma palavra-passe aleatória mostrada na primeira inicialização do contentor.
- Se não for definido e o utilizador *admin* existir, nenhuma ação será executada.
- Se definido, o utilizador *admin* é ajustado em cada inicialização do contentor para corresponder a `WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME` e `WEBLATE_ADMIN_EMAIL`.

> **Aviso:** Pode ser um risco de segurança armazenar a palavra-passe no ficheiro de configuração. Considere usar essa variável apenas para configuração inicial (ou deixe o Weblate gerar uma palavra-passe aleatória na inicialização) ou para recuperação de palavra-passe.

Veja também:

Autenticação como administrador, *WEBLATE_ADMIN_PASSWORD*, *WEBLATE_ADMIN_PASSWORD_FILE*, *WEBLATE_ADMIN_NAME*, *WEBLATE_ADMIN_EMAIL*

WEBLATE_ADMIN_PASSWORD_FILE

Define o caminho para um ficheiro que contém a palavra-passe para o utilizador *admin*.

Veja também:

WEBLATE_ADMIN_PASSWORD

WEBLATE_SERVER_EMAIL

O endereço de e-mail a partir do qual as mensagens de erro são enviadas.

Veja também:

SERVER_EMAIL, *Configurar envio de e-mail*

WEBLATE_DEFAULT_FROM_EMAIL

Configura o endereço para e-mails de saída.

Veja também:

DEFAULT_FROM_EMAIL, *Configurar envio de e-mail*

WEBLATE_CONTACT_FORM

Configura o comportamento do formulário de contato, veja *CONTACT_FORM*.

WEBLATE_ALLOWED_HOSTS

Configura os nomes de host HTTP permitidos a usar *ALLOWED_HOSTS*.

O padrão é `` * `` que permite todos os nomes de host.

Exemplo:

```
environment:
  WEBLATE_ALLOWED_HOSTS: weblate.example.com,example.com
```

Veja também:

ALLOWED_HOSTS, *Configuração de hosts permitidos*, *Definir domínio correto do site*

WEBLATE_REGISTRATION_OPEN

Configura se os registos são abertos a alternar *REGISTRATION_OPEN*.

Exemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Configura quais métodos de autenticação podem ser usados para criar uma nova conta via *REGISTRATION_ALLOW_BACKENDS*.

Exemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
  WEBLATE_REGISTRATION_ALLOW_BACKENDS: azuread-oauth2,azuread-tenant-
↪oauth2
```

WEBLATE_TIME_ZONE

Configura o fuso horário usado no Weblate, veja `django: TIME_ZONE`.

Nota: Para alterar o fuso horário do próprio contentor do Docker, use a variável de ambiente `` TZ``.

Exemplo:

```
environment:
  WEBLATE_TIME_ZONE: Europe/Prague
```

WEBLATE_ENABLE_HTTPS

Faz com que o Weblate presuma que é operado por trás de um proxy HTTPS reverso, faz com que o Weblate use HTTPS em e-mail e ligações de API ou defina sinalizadores seguros em cookies.

Dica: Por favor, consulte a documentação de *ENABLE_HTTPS* para possíveis advertências.

Nota: Isso não faz com que o contentor Weblate aceite conexões HTTPS, precisa configurar isso também, consulte *Contentor Docker com suporte a HTTPS* para exemplos.

Exemplo:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Veja também:

ENABLE_HTTPS *Definir domínio correto do site*, *WEBLATE_SECURE_PROXY_SSL_HEADER*

WEBLATE_INTERLEDGER_PAYMENT_POINTERS

Novo na versão 4.12.1.

Permite que o Weblate defina o campo *meta[nome=monetização]* no cabeçalho do documento. Se vários forem especificados, escolhe um aleatoriamente.

Veja também:

INTERLEDGER_PAYMENT_POINTERS

WEBLATE_IP_PROXY_HEADER

Permite que o Weblate obtenha o endereço IP de qualquer cabeçalho HTTP fornecido. Use isso ao usar um proxy reverso na frente do contentor Weblate.

Ativa *IP_BEHIND_REVERSE_PROXY* e define *IP_PROXY_HEADER*.

Nota: O formato deve estar de acordo com as expectativas do Django. O Django transforma nomes de cabeçalho HTTP brutos da seguinte forma:

- converte todos os caracteres em maiúsculas
- substitui todos os hifenes por sublinhados
- prefixa o prefixo `HTTP_`

Portanto, `X-Forwarded-For` seria mapeado para `HTTP_X_FORWARDED_FOR`.

Exemplo:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Uma tupla que representa uma combinação de cabeçalho/valor HTTP que significa que uma solicitação é segura. Isso é necessário quando o Weblate está a ser executado por trás de um proxy reverso a fazer a terminação SSL que não passa cabeçalhos HTTPS padrão.

Exemplo:

```
environment:
  WEBLATE_SECURE_PROXY_SSL_HEADER: HTTP_X_FORWARDED_PROTO,https
```

Veja também:

SECURE_PROXY_SSL_HEADER

WEBLATE_REQUIRE_LOGIN

Ativa *REQUIRE_LOGIN* para impor autenticação em todo o Weblate.

Exemplo:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Adiciona exceções de URL para autenticação necessária para toda a instalação do Weblate a usar *LOGIN_REQUIRED_URLS_EXCEPTIONS*.

Pode substituir configurações inteiras ou modificar o valor padrão a usar as variáveis ADD e REMOVE.

WEBLATE_GOOGLE_ANALYTICS_ID

Configura o ID para o Google Analytics a alterar *GOOGLE_ANALYTICS_ID*.

WEBLATE_GITHUB_USERNAME

Configura o nome de utilizador do GitHub para pull requests do GitHub a alterar *GITHUB_USERNAME*.

Veja também:

GitHub pull requests

WEBLATE_GITHUB_TOKEN

Novo na versão 4.3.

Configura o token de acesso pessoal do GitHub para pull requests do GitHub via API a alterar *GITHUB_TOKEN*.

Veja também:

GitHub pull requests

WEBLATE_GITLAB_USERNAME

Configura o nome de utilizador do GitLab para merge requests do GitLab a alterar *GITLAB_USERNAME*

Veja também:

Solicitação de mesclagem do GitLab

WEBLATE_GITLAB_TOKEN

Configura o token de acesso pessoal do GitLab para merge requests do GitLab via API a alterar *GITLAB_TOKEN*

Veja também:

Solicitação de mesclagem do GitLab

WEBLATE_PAGURE_USERNAME

Configura o nome de utilizador do Pagure para merge requests do Pagure a alterar *PAGURE_USERNAME*

Veja também:

Pagure merge requests

WEBLATE_PAGURE_TOKEN

Configura o token de acesso pessoal do Pagure para merge requests do Pagure via API a alterar *PAGURE_TOKEN*

Veja também:

Pagure merge requests

WEBLATE_DEFAULT_PULL_MESSAGE

Configura o título e a mensagem padrão para solicitações de pull via API alterando *GITHUB_TOKEN*

Veja também:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

Configura a política de simplificação de idioma, veja *SIMPLIFY_LANGUAGES*.

WEBLATE_DEFAULT_ACCESS_CONTROL

Configura o padrão *Controlo de acesso* para novos projetos, veja *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configura o valor padrão para *Acesso restrito* para novos componentes, veja *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configura o valor padrão para *Permitir propagação da tradução* para novos componentes, veja *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Configura *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Configura *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

Configura *DEFAULT_SHARED_TM*.

WEBLATE_AKISMET_API_KEY

Configura a chave API do Akismet, veja *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Configura a assinatura GPG de commits, veja *WEBLATE_GPG_IDENTITY*.

Veja também:

Assinando commits do Git com GnuPG

WEBLATE_URL_PREFIX

Configura o prefixo da URL onde o Weblate está a ser executado, veja *URL_PREFIX*.

WEBLATE_SILENCED_SYSTEM_CHECKS

Configura verificações que não deseja que sejam mostradas, veja `django: SILENCED_SYSTEM_CHECKS`.

WEBLATE_CSP_SCRIPT_SRC

WEBLATE_CSP_IMG_SRC

WEBLATE_CSP_CONNECT_SRC

WEBLATE_CSP_STYLE_SRC

WEBLATE_CSP_FONT_SRC

Permite personalizar o cabeçalho HTTP `Content-Security-Policy`.

Veja também:

Política de segurança de conteúdo, *CSP_SCRIPT_SRC*, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

WEBLATE_LICENSE_FILTER

Configura *LICENSE_FILTER*.

WEBLATE_LICENSE_REQUIRED

Configura *LICENSE_REQUIRED*

WEBLATE_WEBSITE_REQUIRED

Configura *WEBSITE_REQUIRED*

WEBLATE_HIDE_VERSION

Configura *HIDE_VERSION*.

WEBLATE_BASIC_LANGUAGES

Configura *BASIC_LANGUAGES*.

WEBLATE_DEFAULT_AUTO_WATCH

Configura *DEFAULT_AUTO_WATCH*.

WEBLATE_RATELIMIT_ATTEMPTS

WEBLATE_RATELIMIT_LOCKOUT

WEBLATE_RATELIMIT_WINDOW

Novo na versão 4.6.

Configura o limitador de taxa.

Dica: Pode definir a configuração para qualquer escopo do limitador de taxa. Para fazer isso, adicione o prefixo `WEBLATE_` a qualquer uma das configurações descritas em *Limitação de taxa*.

Veja também:

Limitação de taxa, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

WEBLATE_API_RATELIMIT_ANON

WEBLATE_API_RATELIMIT_USER

Novo na versão 4.11.

Configures API rate limiting. Defaults to `100/day` for anonymous and `5000/hour` for authenticated users.

Veja também:

Limitação de taxa da API

WEBLATE_ENABLE_HOOKS

Novo na versão 4.13.

Configures *ENABLE_HOOKS*.

WEBLATE_ENABLE_AVATARS

Novo na versão 4.6.1.

Configura *ENABLE_AVATARS*.

WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Novo na versão 4.9.

Configura *LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH*.

WEBLATE_SSH_EXTRA_ARGS

Novo na versão 4.9.

Configura *SSH_EXTRA_ARGS*.

WEBLATE_BORG_EXTRA_ARGS

Novo na versão 4.9.

Configura *BORG_EXTRA_ARGS*.

WEBLATE_ENABLE_SHARING

Novo na versão 4.14.1.

Configures *ENABLE_SHARING*.

Automatic suggestion settings

Alterado na versão 4.13: Automatic suggestion services are now configured in the user interface, see *Configuring automatic suggestions*.

As variáveis de ambiente atuais são importadas durante a migração ao Weblate 4.13, mas alterá-las não surtirá nenhum efeito.

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_BIND_PASSWORD_FILE

Path to the file containing the LDAP server bind password.

Veja também:

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)
```

Veja também:

Autenticação por LDAP

GitHub

WEBLATE_SOCIAL_AUTH_GITHUB_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET

WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID

Ativa *Autenticação por GitHub*.

Bitbucket

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY

WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET

Ativa *Autenticação por Bitbucket*.

Facebook

WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY

WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET

Ativa *OAuth 2 do Facebook*.

Google

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS

Ativa *OAuth 2 do Google*.

GitLab

WEBLATE_SOCIAL_AUTH_GITLAB_KEY

WEBLATE_SOCIAL_AUTH_GITLAB_SECRET

WEBLATE_SOCIAL_AUTH_GITLAB_API_URL

Ativa *OAuth 2 do GitLab*.

Active Directory do Azure

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET

Ativa autenticação por Active Directory do Azure, veja *Active Directory do Microsoft Azure*.

Active Directory do Azure com suporte a Tenant

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET

WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID

Ativa autenticação por Active Directory do Azure com suporte a Tenant, veja *Active Directory do Microsoft Azure*.

Keycloak

WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET

WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM

WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL

WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE

WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE

Ativa autenticação com Keycloak, veja a documentação.

Fornecedores Linux

Pode ativar a autenticação a usar serviços de autenticação de fornecedores Linux, a definir as seguintes variáveis para qualquer valor.

WEBLATE_SOCIAL_AUTH_FEDORA

WEBLATE_SOCIAL_AUTH_OPENSUSE

WEBLATE_SOCIAL_AUTH_UBUNTU

Slack

WEBLATE_SOCIAL_AUTH_SLACK_KEY

SOCIAL_AUTH_SLACK_SECRET

Ativa a autenticação Slack, veja *Slack*.

OpenID Connect

Novo na versão 4.13-1.

WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT

WEBLATE_SOCIAL_AUTH_OIDC_KEY

WEBLATE_SOCIAL_AUTH_OIDC_SECRET

WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY

Configures generic OpenID Connect intergration.

Veja também:

OIDC (OpenID Connect)

SAML

Chaves SAML autoassinadas são geradas automaticamente na primeira inicialização do contentor. Caso queira usar chaves próprias, ponha o certificado e a chave privada em `/app/data/ssl/saml.crt` e `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

WEBLATE_SAML_IDP_IMAGE

WEBLATE_SAML_IDP_TITLE

Configurações do provedor de identidade SAML, consulte *Autenticação por SAML*.

Outras configurações de autenticação

WEBLATE_NO_EMAIL_AUTH

Disables e-mail authentication when set to any value. See *Desativar autenticação por palavra-passe*.

Configuração de banco de dados PostgreSQL

O banco de dados é criado por `docker-compose.yml`, então essas configurações afetam os contentores Weblate e PostgreSQL.

Veja também:

Configuração de banco de dados para o Weblate

POSTGRES_PASSWORD

Palavra-passe do PostgreSQL.

POSTGRES_PASSWORD_FILE

Caminho para o ficheiro que contém a palavra-passe do PostgreSQL. Use como uma alternativa para POSTGRES_PASSWORD.

POSTGRES_USER

Nome de utilizador do PostgreSQL.

POSTGRES_DATABASE

Nome do banco de dados PostgreSQL.

POSTGRES_HOST

Nome de host ou endereço IP do servidor PostgreSQL. O padrão é `database`.

POSTGRES_PORT

Porta do servidor PostgreSQL. O padrão é nenhum (usa o valor padrão).

POSTGRES_SSL_MODE

Configura como o PostgreSQL lida com SSL em conexão com o servidor, para as opções possíveis, consulte SSL Mode Descriptions

POSTGRES_ALTER_ROLE

Configura o nome da função para alterar durante as migrações, consulte *Configurar Weblate para usar PostgreSQL*.

POSTGRES_CONN_MAX_AGE

Novo na versão 4.8.1.

O tempo de vida de uma conexão de banco de dados, como um número inteiro de segundos. Use 0 para fechar as conexões do banco de dados no final de cada requisição (este é o comportamento predefinido).

Ativar a persistência da conexão normalmente causará uma conexão mais aberta com o banco de dados. Por favor, ajuste a sua configuração do banco de dados antes de ativar.

Exemplo de configuração:

```
environment:
    POSTGRES_CONN_MAX_AGE: 3600
```

Veja também:

`CONN_MAX_AGE`, Persistent connections

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

Novo na versão 4.9.1.

Desativa os cursores do lado do servidor no banco de dados. Isso é necessário em algumas configurações do **pgbouncer**.

Exemplo de configuração:

```
environment:
    POSTGRES_DISABLE_SERVER_SIDE_CURSORS: 1
```

Veja também:

`DISABLE_SERVER_SIDE_CURSORS`, Transaction pooling and server-side cursors

Configurações de backup de base de dados

Veja também:

Dados despejados para backups

WEBLATE_DATABASE_BACKUP

Configura o despejo diário do banco de dados a usar *`DATABASE_BACKUP`*. O padrão é `plain`.

Configuração do servidor de cache

O uso do Redis é altamente recomendado pelo Weblate e deve fornecer uma instância do Redis ao executar o Weblate no Docker.

Veja também:

Ativar o cache

REDIS_HOST

O nome de host ou endereço IP do servidor Redis. O padrão é `cache`.

REDIS_PORT

A porta do servidor Redis. O padrão é `6379`.

REDIS_DB

O número do banco de dados Redis, o padrão é `1`.

REDIS_PASSWORD

A palavra-passe do servidor Redis, não usada por padrão.

REDIS_PASSWORD_FILE

Path to the file containing the Redis server password.

Veja também:

REDIS_PASSWORD

REDIS_TLS

Ativa o uso de SSL para conexão Redis.

REDIS_VERIFY_SSL

Pode ser usado para desativar a verificação de certificado SSL para conexão Redis.

Configuração do servidor de e-mail

Para fazer com que o e-mail de saída funcione, precisa fornecer um servidor de e-mail.

Exemplo de configuração TLS:

```
environment:
    WEBLATE_EMAIL_HOST: smtp.example.com
    WEBLATE_EMAIL_HOST_USER: user
    WEBLATE_EMAIL_HOST_PASSWORD: pass
```

Exemplo de configuração SSL:

```
environment:
    WEBLATE_EMAIL_HOST: smtp.example.com
    WEBLATE_EMAIL_PORT: 465
    WEBLATE_EMAIL_HOST_USER: user
    WEBLATE_EMAIL_HOST_PASSWORD: pass
    WEBLATE_EMAIL_USE_TLS: 0
    WEBLATE_EMAIL_USE_SSL: 1
```

Veja também:

Configuração de e-mail de saída

WEBLATE_EMAIL_HOST

Nome de host ou endereço IP do servidor de correio.

Veja também:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, *WEBLATE_EMAIL_USE_TLS*, EMAIL_HOST

WEBLATE_EMAIL_PORT

Porta do servidor de correio, o padrão é 25.

Veja também:

EMAIL_PORT

WEBLATE_EMAIL_HOST_USER

Utilizador da autenticação por e-mail.

Veja também:

EMAIL_HOST_USER

WEBLATE_EMAIL_HOST_PASSWORD

Palavra-passe da autenticação por e-mail.

Veja também:

EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Caminho para o ficheiro que contém a palavra-passe da autenticação por e-mail.

Veja também:

WEBLATE_EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_USE_SSL

Se deve usar uma conexão TLS (segura) implícita ao falar com o servidor SMTP. Na maioria das documentações de e-mail, esse tipo de conexão TLS é conhecido como SSL. Geralmente é usado na porta 465. Se estiver a ter problemas, consulte a configuração TLS explícita *WEBLATE_EMAIL_USE_TLS*.

Alterado na versão 4.11: O suporte a SSL/TLS é ativado automaticamente com base em *WEBLATE_EMAIL_PORT*.

Veja também:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_TLS*, EMAIL_USE_SSL

WEBLATE_EMAIL_USE_TLS

Se deve usar uma conexão TLS (segura) ao falar com o servidor SMTP. Isso é usado para conexões TLS explícitas, geralmente na porta 587 ou 25. Se estiver a ter conexões travadas, consulte a configuração TLS implícita *WEBLATE_EMAIL_USE_SSL*.

Alterado na versão 4.11: O suporte a SSL/TLS é ativado automaticamente com base em *WEBLATE_EMAIL_PORT*.

Veja também:

WEBLATE_EMAIL_PORT, *WEBLATE_EMAIL_USE_SSL*, EMAIL_USE_TLS

WEBLATE_EMAIL_BACKEND

Configura o back-end do Django para usar no envio de e-mails.

Veja também:

Configurar envio de e-mail, EMAIL_BACKEND

WEBLATE_AUTO_UPDATE

Configura se e como o Weblate deve atualizar os repositórios.

Veja também:

AUTO_UPDATE

Nota: Esta é uma configuração booleana (use `"true"` ou `"false"`).

Integração do site

WEBLATE_GET_HELP_URL

Configura *GET_HELP_URL*.

WEBLATE_STATUS_URL

Configura *STATUS_URL*.

WEBLATE_LEGAL_URL

Configura *LEGAL_URL*.

WEBLATE_PRIVACY_URL

Configura *PRIVACY_URL*.

Relatório de erro

É recomendado coletar erros da instalação sistematicamente, veja *Coletando relatórios de erros*.

Para ativar o suporte para Rollbar, defina o seguinte:

ROLLBAR_KEY

O seu token de acesso ao servidor de postagem Rollbar.

ROLLBAR_ENVIRONMENT

O seu ambiente Rollbar, o padrão é `production`.

Para ativar o suporte para Sentry, defina o seguinte:

SENTRY_DSN

O seu DSN no Sentry.

SENTRY_ENVIRONMENT

O seu ambiente no Sentry (opcional).

CDN de localização

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

Novo na versão 4.2.1.

Configuração para *CDN de localização JavaScript*.

O *WEBLATE_LOCALIZE_CDN_PATH* é o caminho dentro do contentor. Ele deve ser armazenado no volume persistente e não no armazenamento temporário.

Uma das possibilidades é armazenar isso dentro do diretório de dados do Weblate:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

Nota: É responsável por configurar o serviço dos ficheiros gerados pelo Weblate, ele só armazena os ficheiros no local configurado.

Veja também:

weblate-cdn, *LOCALIZE_CDN_URL*, *LOCALIZE_CDN_PATH*

Alterar aplicações, verificações, extensões ou correções automáticas ativados

Novo na versão 3.8-5.

A configuração embutida de verificações, extensões ou correções automática ativados pode ser ajustada pelas seguintes variáveis:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Exemplo:

```
environment:
  WEBLATE_REMOVE_AUTOFIX: weblate.trans.autofixes.whitespace.
↪SameBookendingWhitespace
  WEBLATE_ADD_ADDONS: customize.addons.MyAddon,customize.addons.OtherAddon
```

Veja também:

CHECK_LIST, *AUTOFIX_LIST*, *WEBLATE_ADDONS*, INSTALLED_APPS

Configurações do contentor

WEBLATE_WORKERS

Novo na versão 4.6.1.

Quantidade base de processos de trabalho em execução no contentor. Quando não definido, é determinado automaticamente na inicialização do contentor com base na quantidade de núcleos de CPU disponíveis.

É usado para determinar *CELERY_MAIN_OPTIONS*, *CELERY_NOTIFY_OPTIONS*, *CELERY_MEMORY_OPTIONS*, *CELERY_TRANSLATE_OPTIONS*, *CELERY_BACKUP_OPTIONS*, *CELERY_BEAT_OPTIONS* e *WEB_WORKERS*. Pode usar essas configurações para fazer o ajuste fino.

CELERY_MAIN_OPTIONS

CELERY_NOTIFY_OPTIONS

CELERY_MEMORY_OPTIONS

CELERY_TRANSLATE_OPTIONS

CELERY_BACKUP_OPTIONS

CELERY_BEAT_OPTIONS

Essas variáveis permitem que ajuste as opções do worker do Celery. Pode ser útil ajustar a simultaneidade (`--concurrency 16`) ou usar diferentes implementações de pool (`--pool=gevent`).

Por padrão, a quantidade de workers simultâneos é baseado em *WEBLATE_WORKERS*.

Exemplo:

```
environment:
  CELERY_MAIN_OPTIONS: --concurrency 16
```

Veja também:

Opções de worker do Celery, *Tarefas de fundo a usar o Celery*

WEB_WORKERS

Configura quantos workers uWSGI devem ser executados.

O padrão é `WEBLATE_WORKERS`.

Exemplo:

```
environment:
  WEB_WORKERS: 32
```

WEBLATE_SERVICE

Define quais serviços devem ser executados dentro do contentor. Use isto para *Dimensionando horizontalmente*.

Os seguintes serviços são definidos:

celery-beat
Agendador de tarefas do Celery, apenas uma instância deve estar em execução. Este contentor também é responsável pelas migrações da estrutura do banco de dados e deve ser iniciado antes dos demais.

celery-backup
Worker do Celery para backups, apenas uma instância deve estar em execução.

celery-celery
Worker genérico do Celery.

celery-memory
Worker do Celery para memória de tradução.

celery-notify
Worker do Celery para notificações.

celery-translate
Worker do Celery para tradução automática.

web
Servidor web.

Volumes de contentor Docker

Há dois volumes (dados e cacho) exportados pelo contentor Weblate. Os outros contentores de serviço (PostgreSQL ou Redis) também têm os volumes de dados deles, mas eles não são cobertos por este documento.

O volume de dados é usado para armazenar dados persistentes do Weblate, como repositórios clonados ou para personalizar a instalação do Weblate.

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/` (o caminho consiste no nome de seu diretório docker-compose, contentor e nomes de volume). No contentor, ele é montado como `/app/data`.

O volume do cache é montado como `/app/cache` e é usado para armazenar ficheiros estáticos. O conteúdo deles é recriado na inicialização do contentor e o volume pode ser montado a usar um sistema de ficheiros efêmero como *tmpfs*.

Ao criar os volumes manualmente, os diretórios devem pertencer ao UID 1000, pois é o utilizador usado dentro do contentor.

Veja também:

Documentação sobre volumes do Docker

Personalização adicional da configuração

Pode personalizar ainda mais a instalação do Weblate no volume de dados, veja *Volumes de contentor Docker*.

Ficheiros de configuração personalizados

Também pode sobrescrever a configuração em `/app/data/settings-override.py` (veja *Volumes de contentor Docker*). Isso é executado no final das configurações embutidas, depois que todas as configurações de ambiente são carregadas e pode ajustá-las ou substituí-las.

Substituindo o logotipo e outros ficheiros estáticos

Novo na versão 3.8-5.

Os ficheiros estáticos que vêm com Weblate podem ser sobrescritos a colocar em `/app/data/python/customize/static` (veja *Volumes de contentor Docker*). Por exemplo, criar `/app/data/python/customize/static/favicon.ico` substituirá o favicon.

Dica: Os ficheiros são copiados para o local correspondente na inicialização do contentor, portanto, é necessário reiniciar o Weblate após alterar o conteúdo do volume.

Essa abordagem também pode ser usada para substituir os modelos Weblate. Por exemplo documentos *legais* podem ser postos em `/app/data/python/customize/templates/legal/documents`.

Como alternativa, também pode incluir o próprio módulo (veja ../ customize) e adicioná-lo como um volume separado ao contentor do Docker, por exemplo:

```
weblate:
  volumes:
    - weblate-data:/app/data
    - ./weblate_customization/weblate_customization:/app/data/python/weblate_
↪customization
  environment:
    WEBLATE_ADD_APPS: weblate_customization
```

Adicionando os seus próprios módulos Python

Novo na versão 3.8-5.

Pode pôr os próprios módulos Python em `/app/data/python/` (veja *Volumes de contentor Docker*) e eles podem ser carregados pelo Weblate, provavelmente a usar docker-custom -config.

Veja também:

Personalizar o Weblate

Configuring PostgreSQL server

O contentor PostgtreSQL usa a configuração padrão do PostgreSQL e não utilizará efetivamente os núcleos de CPU ou memória dele. Recomenda-se personalizar a configuração para melhorar o desempenho.

A configuração pode ser ajustada conforme descrito em *Database Configuration* em https://hub.docker.com/_/postgres. A configuração correspondente ao seu ambiente pode ser gerada usando https://pgtune.leopard.in.ua/.

Instalando no Debian e Ubuntu

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
apt install -y \
   libxml2-dev libxslt-dev libfreetype6-dev libjpeg-dev libz-dev libyaml-dev \
   libffi-dev libcairo-dev gir1.2-pango-1.0 libgirepository1.0-dev \
   libacl1-dev libssl-dev libpq-dev libjpeg62-turbo-dev build-essential \
   python3-gdbm python3-dev python3-pip python3-virtualenv virtualenv git
```

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
apt install -y \
   tesseract-ocr libtesseract-dev libleptonica-dev \
   libldap2-dev libldap-common libsasl2-dev \
   libxmlsec1-dev
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação local:

```
# Web server option 1: NGINX and uWSGI
apt install -y nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
apt install -y apache2 libapache2-mod-wsgi-py3

# Caching backend: Redis
apt install -y redis-server

# Database server: PostgreSQL
apt install -y postgresql postgresql-contrib

# SMTP server
apt install -y exim4
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique virtualenv venv: user_guide.

1. Crie o virtualenv para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Ative o virtualenv para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale o Weblate incluindo todas as dependências opcionais:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Confira *Dependências opcionais* para o ajuste fino das dependências opcionais.

   Nota: Em algumas distribuições Linux, a execução do Weblate falha com erro de libffi:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   Isso é causado pela incompatibilidade de pacotes binários distribuídos via PyPI com a distribuição. Para resolver isso, precisa reconstruir o pacote no seu sistema:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Configurando o Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas vai querer fazer mais mudanças para configuração de produção, veja *Ajustar a configuração*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuração de banco de dados para o Weblate* for a production-ready setup):

```
weblate migrate
```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect the static files for your web server (see *Executar o servidor* and *Servir ficheiros estáticos*):

```
weblate collectstatic
```

6. Compress the JavaScript and CSS files (optional, see *Comprimir os ativos do cliente*):

```
weblate compress
```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tarefas de fundo a usar o Celery* has more info:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Start the development server (*Executar o servidor* details a production setup):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- Agora pode executar comandos Weblate a usar o comando **`weblate`** quando virtualenv do Weblate está ativo, veja *Comandos de gerência*.
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com a sua instalação na URL `/manage/performance/` (veja *Interface de gestão*) ou a usar **`weblate check --deploy`**, veja *Configuração de produção*.

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja *Configuração de projeto* para mais detalhes.

   Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja *Configuração de componente* para mais detalhes.

   The important fields here are: *Nome do componente*, *Repositório do código-fonte*, and *Máscara de ficheiros* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Recurso de cadeias de Android*, *Cadeias de Apple iOS*, *Propriedades Java*, *Formato stringsdict* or *Formato Fluent*, see *Formatos de ficheiros suportados* for more details.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando no SUSE e openSUSE

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
zypper install \
   libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
   libyaml-devel libffi-devel cairo-devel pango-devel \
   gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
   python3-devel git
```

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
zypper install tesseract-ocr tesseract-devel leptonica-devel
zypper install libldap2-devel libsasl2-devel
zypper install libxmlsec1-devel
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação local:

```
# Web server option 1: NGINX and uWSGI
zypper install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
zypper install apache2 apache2-mod_wsgi

# Caching backend: Redis
zypper install redis-server

# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique virtualenv venv: user_guide.

1. Crie o virtualenv para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Ative o virtualenv para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale o Weblate incluindo todas as dependências opcionais:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Confira *Dependências opcionais* para o ajuste fino das dependências opcionais.

   Nota: Em algumas distribuições Linux, a execução do Weblate falha com erro de libffi:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   Isso é causado pela incompatibilidade de pacotes binários distribuídos via PyPI com a distribuição. Para resolver isso, precisa reconstruir o pacote no seu sistema:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Configurando o Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas vai querer fazer mais mudanças para configuração de produção, veja *Ajustar a configuração*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuração de banco de dados para o Weblate* for a production-ready setup):

```
weblate migrate
```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect the static files for your web server (see *Executar o servidor* and *Servir ficheiros estáticos*):

```
weblate collectstatic
```

6. Compress the JavaScript and CSS files (optional, see *Comprimir os ativos do cliente*):

```
weblate compress
```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tarefas de fundo a usar o Celery* has more info:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Start the development server (*Executar o servidor* details a production setup):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- Agora pode executar comandos Weblate a usar o comando **`weblate`** quando virtualenv do Weblate está ativo, veja *Comandos de gerência*.
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com a sua instalação na URL `/manage/performance/` (veja *Interface de gestão*) ou a usar **`weblate check --deploy`**, veja *Configuração de produção*.

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja *Configuração de projeto* para mais detalhes.

   Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja *Configuração de componente* para mais detalhes.

   The important fields here are: *Nome do componente*, *Repositório do código-fonte*, and *Máscara de ficheiros* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Recurso de cadeias de Android*, *Cadeias de Apple iOS*, *Propriedades Java*, *Formato stringsdict* or *Formato Fluent*, see *Formatos de ficheiros suportados* for more details.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando no RedHat, Fedora e CentOS

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
dnf install \
   libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
   libyaml-devel libffi-devel cairo-devel pango-devel \
   gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
   python3-devel git
```

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
dnf install tesseract-langpack-eng tesseract-devel leptonica-devel
dnf install libldap2-devel libsasl2-devel
dnf install libxmlsec1-devel
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação local:

```
# Web server option 1: NGINX and uWSGI
dnf install nginx uwsgi uwsgi-plugin-python3

# Web server option 2: Apache with ``mod_wsgi``
dnf install apache2 apache2-mod_wsgi

# Caching backend: Redis
dnf install redis

# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique virtualenv venv: user_guide.

1. Crie o virtualenv para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Ative o virtualenv para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale o Weblate incluindo todas as dependências opcionais:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Confira *Dependências opcionais* para o ajuste fino das dependências opcionais.

   Nota: Em algumas distribuições Linux, a execução do Weblate falha com erro de libffi:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   Isso é causado pela incompatibilidade de pacotes binários distribuídos via PyPI com a distribuição. Para resolver isso, precisa reconstruir o pacote no seu sistema:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Configurando o Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas vai querer fazer mais mudanças para configuração de produção, veja *Ajustar a configuração*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuração de banco de dados para o Weblate* for a production-ready setup):

```
weblate migrate
```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect the static files for your web server (see *Executar o servidor* and *Servir ficheiros estáticos*):

```
weblate collectstatic
```

6. Compress the JavaScript and CSS files (optional, see *Comprimir os ativos do cliente*):

```
weblate compress
```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tarefas de fundo a usar o Celery* has more info:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Start the development server (*Executar o servidor* details a production setup):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- Agora pode executar comandos Weblate a usar o comando **`weblate`** quando virtualenv do Weblate está ativo, veja *Comandos de gerência*.
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com a sua instalação na URL `/manage/performance/` (veja *Interface de gestão*) ou a usar **`weblate check --deploy`**, veja *Configuração de produção*.

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja *Configuração de projeto* para mais detalhes.

   Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja *Configuração de componente* para mais detalhes.

   The important fields here are: *Nome do componente*, *Repositório do código-fonte*, and *Máscara de ficheiros* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Recurso de cadeias de Android*, *Cadeias de Apple iOS*, *Propriedades Java*, *Formato stringsdict* or *Formato Fluent*, see *Formatos de ficheiros suportados* for more details.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando no macOS

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate num único host (Weblate, banco de dados e servidor web):

- 2 GB de RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de ficheiros, banco de dados e Weblate).

Muitos utilizadores simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter o tamanho mínimo deles a fazer clones rasos.

Nota: Os requisitos reais para a sua instalação do Weblate variam fortemente com base no tamanho das traduções geridas nele.

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
pip install virtualenv
```

Certifique-se de que o pip será capaz de encontrar a versão do `libffi` fornecida pelo homebrew – isso será necessário durante a etapa de compilação da instalação.

```
export PKG_CONFIG_PATH="/usr/local/opt/libffi/lib/pkgconfig"
```

Instale as dependências opcionais desejadas, a depender dos recursos que pretende usar (veja *Dependências opcionais*):

```
brew install tesseract
```

Opcionalmente, instale o software para executar o servidor de produção, veja *Executar o servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo a usar o Celery*. Dependendo do tamanho da sua instalação, poderá executar esses componentes em servidores dedicados.

As instruções de instalação local:

```
# Web server option 1: NGINX and uWSGI
brew install nginx uwsgi

# Web server option 2: Apache with ``mod_wsgi``
brew install httpd

# Caching backend: Redis
brew install redis

# Database server: PostgreSQL
brew install postgresql
```

Módulos do Python

Dica: Estamos a usar o virtualenv para instalar o Weblate num ambiente separado do seu sistema. Se não estiver familiarizado com ele, verifique virtualenv venv: user_guide.

1. Crie o virtualenv para Weblate:

   ```
   virtualenv ~/weblate-env
   ```

2. Ative o virtualenv para Weblate:

   ```
   . ~/weblate-env/bin/activate
   ```

3. Instale o Weblate incluindo todas as dependências opcionais:

   ```
   # Install Weblate with all optional dependencies
   pip install "Weblate[all]"
   ```

   Confira *Dependências opcionais* para o ajuste fino das dependências opcionais.

   Nota: Em algumas distribuições Linux, a execução do Weblate falha com erro de libffi:

   ```
   ffi_prep_closure(): bad user_data (it seems that the version of the libffi␣
   ↪library seen at runtime is different from the 'ffi.h' file seen at compile-
   ↪time)
   ```

   Isso é causado pela incompatibilidade de pacotes binários distribuídos via PyPI com a distribuição. Para resolver isso, precisa reconstruir o pacote no seu sistema:

   ```
   pip install --force-reinstall --no-binary :all: cffi
   ```

Configurando o Weblate

Nota: The following assumes the virtualenv used by Weblate is activated (by executing `. ~/weblate-env/bin/activate`). If not, specify the full path to the **`weblate`** command as `~/weblate-env/bin/weblate`.

1. Copie o ficheiro `~/weblate-env/lib/python3.9/site-packages/weblate/settings_example.py` para `~/weblate-env/lib/python3.9/site-packages/weblate/settings.py`.
2. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas vai querer fazer mais mudanças para configuração de produção, veja *Ajustar a configuração*.
3. Create the database and its structure for Weblate (the example settings use PostgreSQL, check *Configuração de banco de dados para o Weblate* for a production-ready setup):

```
weblate migrate
```

4. Create an account for the administrator user and copy its password to the clipboard, and also save it for later use:

```
weblate createadmin
```

5. Collect the static files for your web server (see *Executar o servidor* and *Servir ficheiros estáticos*):

```
weblate collectstatic
```

6. Compress the JavaScript and CSS files (optional, see *Comprimir os ativos do cliente*):

```
weblate compress
```

7. Start the Celery workers. This is not necessary for development purposes, but strongly recommended otherwise. *Tarefas de fundo a usar o Celery* has more info:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

8. Start the development server (*Executar o servidor* details a production setup):

```
weblate runserver
```

Após a instalação

Parabéns, o seu servidor Weblate agora está a funcionar e pode começar a usá-lo.

- Agora pode acessar o Weblate em `http://localhost:8000/`.
- Sign in with admin credentials obtained during installation or register with new users.
- Agora pode executar comandos Weblate a usar o comando **`weblate`** quando virtualenv do Weblate está ativo, veja *Comandos de gerência*.
- Pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com a sua instalação na URL `/manage/performance/` (veja *Interface de gestão*) ou a usar **`weblate check --deploy`**, veja *Configuração de produção*.

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja *Configuração de projeto* para mais detalhes.

   Tudo que precisa especificar aqui é o nome do projeto e seu site.

2. Crie um componente que é o objeto real para tradução – ele aponta ao repositório VCS e seleciona quais ficheiros traduzir. Veja *Configuração de componente* para mais detalhes.

   The important fields here are: *Nome do componente*, *Repositório do código-fonte*, and *Máscara de ficheiros* for finding translatable files. Weblate supports a wide range of formats including *GNU gettext*, *Recurso de cadeias de Android*, *Cadeias de Apple iOS*, *Propriedades Java*, *Formato stringsdict* or *Formato Fluent*, see *Formatos de ficheiros suportados* for more details.

3. Depois que o acima for concluído (pode ser um processo demorado a depender do tamanho do seu repositório VCS e da quantidade de mensagens a serem traduzidas), pode começar a traduzir.

Instalando a partir do código-fonte

1. Please follow the installation instructions for your system first up to installing Weblate:

   - *Instalando no Debian e Ubuntu*
   - *Instalando no SUSE e openSUSE*
   - *Instalando no RedHat, Fedora e CentOS*

2. Pegue o código-fonte mais recente do Weblate a usar Git (ou baixe um tarball e descompacte-o):

   ```
   git clone https://github.com/WeblateOrg/weblate.git weblate-src
   ```

   Alternativamente, pode usar ficheiros lançados. Pode descarregá-los no nosso site <https://weblate.org/>. Essas descarregas são assinadas criptograficamente, consulte *Verificar assinaturas de lançamento*.

3. Instale o código atual do Weblate no virtualenv:

   ```
   . ~/weblate-env/bin/activate
   pip install -e weblate-src
   ```

4. Copie `weblate/settings_example.py` para `weblate/settings.py`.

5. Ajuste os valores no novo ficheiro `settings.py` de acordo com a sua preferência. Vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas vai querer fazer mais mudanças para configuração de produção, veja *Ajustar a configuração*.

6. Crie o banco de dados usado pelo Weblate, veja *Configuração de banco de dados para o Weblate*.

7. Construa tabelas Django, ficheiros estáticos e dados iniciais (veja *Preencher o banco de dados* e *Servir ficheiros estáticos*):

   ```
   weblate migrate
   weblate collectstatic
   weblate compress
   ```

   Nota: Esta etapa deve ser repetida sempre que atualizar o repositório.

Instalando no OpenShift

Com o modelo OpenShift Weblate, pode pôr a sua instância pessoal do Weblate em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão e declarações de volume persistentes são usadas.

Pode encontrar o modelo em <https://github.com/WeblateOrg/openshift/>.

Instalação

Os exemplos a seguir presumem que tem um ambiente OpenShift v3.x funcional, com a ferramenta cliente `oc` instalada. Confira a documentação do OpenShift para obter instruções.

O `template.yml` é adequado para executar todos os componentes em OpenShift. Existe também `template-external-postgresql.yml` que não inicia um servidor PostgreSQL e permite que configure um servidor PostgreSQL externo.

Console web

Copie o conteúdo bruto de template.yml e importe-o ao seu projeto, então use o botão `Create` no console web do OpenShift para criar a sua aplicação. O console web solicitará os valores de todos os parâmetros usados pelo modelo.

CLI

Para enviar o modelo Weblate para a biblioteca de modelos do seu projeto atual, passe o ficheiro `template.yml` com o seguinte comando:

```
$ oc create -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
   -n <PROJECT>
```

O modelo agora está disponível para seleção a usar o console web ou o CLI.

Parâmetros

Os parâmetros que pode substituir estão listados na secção de parâmetros do modelo. Pode listá-los com a CLI a usar o seguinte comando e a especificar o ficheiro a ser usado:

```
$ oc process --parameters -f https://raw.githubusercontent.com/WeblateOrg/
↪openshift/main/template.yml

# If the template is already uploaded
$ oc process --parameters -n <PROJECT> weblate
```

Provisionamento

Também pode usar a CLI para processar modelos e usar a configuração gerada para criar objetos imediatamente.

```
$ oc process -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
    -p APPLICATION_NAME=weblate \
    -p WEBLATE_VERSION=4.3.1-1 \
    -p WEBLATE_SITE_DOMAIN=weblate.app-openshift.example.com \
    -p POSTGRESQL_IMAGE=docker-registry.default.svc:5000/openshift/postgresql:9.6 \
```

(continues on next page)

(continuação da página anterior)

```
    -p REDIS_IMAGE=docker-registry.default.svc:5000/openshift/redis:3.2 \
    | oc create -f
```

A instância do Weblate deve estar disponível após a migração e implantação com sucesso no parâmetro `WEBLATE_SITE_DOMAIN` especificado.

Após a configuração do contentor, pode entrar como utilizador *admin* com a palavra-passe fornecida em `WEBLATE_ADMIN_PASSWORD`, ou uma palavra-passe aleatória gerada na primeira inicialização se não tiver sido definida.

Para redefinir a palavra-passe do *admin*, reinicie o contentor com `WEBLATE_ADMIN_PASSWORD` definido para a nova palavra-passe no respectivo `Secret`.

Eliminar

```
$ oc delete all -l app=<APPLICATION_NAME>
$ oc delete configmap -l app= <APPLICATION_NAME>
$ oc delete secret -l app=<APPLICATION_NAME>
# ATTTENTION! The following command is only optional and will permanently delete␣
↪all of your data.
$ oc delete pvc -l app=<APPLICATION_NAME>

$ oc delete all -l app=weblate \
    && oc delete secret -l app=weblate \
    && oc delete configmap -l app=weblate \
    && oc delete pvc -l app=weblate
```

Configuração

Ao processar o modelo, um respectivo `ConfigMap` será criado e poderá ser usado para personalizar a imagem Weblate. O `ConfigMap` é montado diretamente como variáveis de ambiente e dispara uma nova implantação toda vez que é alterado. Para mais opções de configuração, consulte *Variáveis de ambiente do Docker* para uma lista completa de variáveis de ambiente.

Instalando no Kubernetes

Nota: Este guia procura colaboradores experientes com o Kubernetes para abordar a configuração com mais detalhes.

Com o chart de Kubernetes Helm, pode pôr a sua instância pessoal do Weblate em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão e declarações de volume persistentes são usadas.

Pode encontrar o chart em <https://github.com/WeblateOrg/helm/> e pode ser exibido em <https://artifacthub.io/packages/helm/weblate/weblate>.

Instalação

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

Configuração

Para mais opções de configuração, consulte *Variáveis de ambiente do Docker* para uma lista completa de variáveis de ambiente.

Dependendo da sua configuração e experiência, escolha um método de instalação apropriado para si:

- *Instalando a usar Docker*, recomendado para configurações de produção.
- Instalação virtualenv, recomendada para configurações de produção:
  - *Instalando no Debian e Ubuntu*
  - *Instalando no SUSE e openSUSE*
  - *Instalando no RedHat, Fedora e CentOS*
  - *Instalando no macOS*
- *Instalando a partir do código-fonte*, recomendado para o desenvolvimento.
- *Instalando no OpenShift*
- *Instalando no Kubernetes*

2.1.2 Requisitos de software

Sistema operacional

Weblate é conhecido por funcionar no Linux, FreeBSD e macOS. Outros sistemas como o Unix provavelmente funcionarão também.

O Weblate não é suportado no Windows. Mas ainda pode funcionar e patches são aceitos alegremente.

Outros serviços

Weblate está a usar outros serviços para a operação dele. Precisará pelo menos os seguintes serviços em execução:

- Servidor de banco de dados PostgreSQL, consulte *Configuração de banco de dados para o Weblate*.
- Servidor Redis para cache e fila de tarefas, consulte *Tarefas de fundo a usar o Celery*.
- Servidor SMTP para e-mails de saída, consulte *Configuração de e-mail de saída*.

Dependências Python

Weblate é escrito em Python e tem suporte de Python 3.6 ou mais novo. Pode instalar dependências a usar pip ou dos pacotes de distribuição deles, a lista completa está disponível em `requirements.txt`.

As dependências mais notáveis:

Django
https://www.djangoproject.com/

Celery
https://docs.celeryq.dev/

Translate Toolkit
https://toolkit.translatehouse.org/

translation-finder
https://github.com/WeblateOrg/translation-finder

Python Social Auth
https://python-social-auth.readthedocs.io/

Django REST Framework
https://www.django-rest-framework.org/

Dependências opcionais

Os módulos seguintes são necessários para alguns recursos do Weblate. Pode encontrar todos em `requirements-optional.txt`.

`Mercurial` (opcional para suporte a repositórios *Mercurial*)
https://www.mercurial-scm.org/

`phply` (opcional para *Cadeias de PHP*)
https://github.com/viraptor/phply

`tesserocr` (opcional para OCR em *Contexto visual para cadeias*)
https://github.com/sirfz/tesserocr

`python-akismet` (opcional para *Proteção contra spam*)
https://github.com/Nekmo/python-akismet

`ruamel.yaml` (opcional para *Ficheiros YAML*)
https://pypi.org/project/ruamel.yaml/

`Zeep` (optional for *Microsoft Terminology*)
https://docs.python-zeep.org/

`aeidon` (opcional para *Ficheiros de legenda*)
https://pypi.org/project/aeidon/

`fluent.syntax` (opcional para *Formato Fluent*)
https://projectfluent.org/

Dica: Ao instalar a usar pip, pode especificar diretamente os recursos desejados ao instalar:

```
pip install "Weblate[PHP,Fluent]"
```

Ou pode instalar o Weblate com todos os recursos opcionais:

```
pip install "Weblate[all]"
```

Ou pode instalar o Weblate sem quaisquer recursos opcionais:

```
pip install Weblate
```

Dependências de backend de banco de dados

O Weblate tem suporte de PostgreSQL, MySQL e MariaDB, consulte *Configuração de banco de dados para o Weblate* e a documentação dos backends para mais detalhes.

Outros requisitos do sistema

As dependências seguintes devem ser instaladas no sistema:

`Git`
https://git-scm.com/

Pango, Cairo and related header files and GObject introspection data
https://cairographics.org/, https://pango.gnome.org/, veja *Pango e Cairo*

`git-review` (opcional para suporte de Gerrit)
https://pypi.org/project/git-review/

`git-svn` (opcional para suporte de Subversion)
https://git-scm.com/docs/git-svn

`tesseract` e os dados dele (opcional para OCR de capturas de ecrã)
https://github.com/tesseract-ocr/tesseract

`licensee` (opcional para detetar a licença ao criar o componente)
https://github.com/licensee/licensee

Dependências de tempo de compilação

Para compilar alguns das *dependências Python*, pode precisar instalar as dependências deles. Isso depende de como instala-as, por isso consulte pacotes individuais para obter documentação. Não precisará deles se usar `Wheels` pré-compilado durante a instalação a usar `pip` ou quando usar pacotes de distribuição.

Pango e Cairo

Alterado na versão 3.7.

O Weblate usa Pango e Cairo para renderizar widgets de bitmap (ver promotion) e verificações de renderização (ver *Gerir letras*). Para instalar as ligações Python corretamente para esses, precisa de instalar bibliotecas de sistemas primeiro - precisa tanto do Cairo quanto do Pango, que por sua vez precisam de GLib. Todos esses devem ser instalados com ficheiros de desenvolvimento e dados de introspecção GObject.

2.1.3 Verificar assinaturas de lançamento

Os lançamentos do Weblate são criptograficamente assinados pelo programador que os lançou. Atualmente é Michal Čihař. A impressão digital da chave PGP é:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

e pode obter mais informações de identificação de <https://keybase.io/nijel>.

Deve verificar se a assinatura corresponde ao ficheiro que descarregou. Desta forma, pode ter certeza de que está a usar o mesmo código que foi lançado. Também deve verificar a data da assinatura para ter certeza de que descarregou a versão mais recente.

Cada arquivo é acompanhado de ficheiros `.asc`, os quais contêm a assinatura PGP para ele. Uma vez que tenha ambos na mesma pasta, pode verificar a assinatura:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

Como pode ver, o GPG reclama que não conhece a chave pública. Neste ponto deve fazer um dos seguintes passos:

- Use *wkd* para descarregar a chave:

```
$ gpg --auto-key-locate wkd --locate-keys michal@cihar.com
pub   rsa4096 2009-06-17 [SC]
      63CB1DF1EF12CF2AC0EE5A329C27B31342B7511D
uid           [ultimate] Michal Čihař <michal@cihar.com>
uid           [ultimate] Michal Čihař <nijel@debian.org>
uid           [ultimate] [jpeg image of size 8848]
uid           [ultimate] Michal Čihař (Braiins) <michal.cihar@braiins.cz>
sub   rsa4096 2009-06-17 [E]
sub   rsa4096 2015-09-09 [S]
```

- Descarregue o chaveiro do servidor do Michal e importe-o com:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Descarregue e importe a chave de um dos servidores principais:

```
$ gpg --keyserver hkp://pgp.mit.edu --recv-keys␣
↪87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: key 9C27B31342B7511D: "Michal Čihař <michal@cihar.com>" imported
gpg: Total number processed: 1
gpg:              unchanged: 1
```

Isso vai melhorar a situação um pouco - neste momento pode verificar que a assinatura da chave dada está correta, mas ainda não pode confiar no nome usado na chave:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:                 aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:                 aka "[jpeg image of size 8848]" [ultimate]
gpg:                 aka "Michal Čihař (Braiins) <michal.cihar@braiins.cz>"␣
↪[ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE  5A32 9C27 B313 42B7 511D
```

O problema aqui é que qualquer um poderia emitir a chave com este nome. Precisa garantir que a chave é realmente a propriedade da pessoa mencionada. O Manual de Privacidade do GNU aborda este tópico no capítulo Validating other keys on your public keyring. O método mais confiável é de conhecer o programador pessoalmente e trocar impressões digitais importantes, no entanto, também pode confiar na rede de confiança. Dessa forma, pode confiar na chave transitivamente por assinaturas de outros, que conheceram o programador pessoalmente.

Uma vez que a chave seja confiável, o aviso não ocorrerá:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:                 aka "Michal Čihař <nijel@debian.org>" [ultimate]
```

(continues on next page)

(continuação da página anterior)

```
gpg:                 aka "[jpeg image of size 8848]" [ultimate]
gpg:                 aka "Michal Čihař (Braiins) <michal.cihar@braiins.cz>"␣
↪[ultimate]
```

Se a assinatura for inválida (o ficheiro foi alterado), obteria um erro claro, independentemente do fato de que a chave é confiável ou não:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: Signature made Sun Mar  3 16:43:15 2019 CET
gpg:                using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: BAD signature from "Michal Čihař <michal@cihar.com>" [ultimate]
```

2.1.4 Permissões do sistema de ficheiros

O processo Weblate precisa ser capaz de ler e escrever para o diretório onde mantém os dados – `DATA_DIR`. Todos os ficheiros dentro deste diretório devem ser de propriedade e graváveis pelo utilizador que executa todos os processos do Weblate (geralmente WSGI e Celery, veja *Executar o servidor* e *Tarefas de fundo a usar o Celery*).

A configuração predefinida põe-os na mesma árvore que as fontes do Weblate, no entanto, pode preferir movê-los para um local melhor, como `/var/lib/weblate`.

O Weblate tenta criar esses diretórios automaticamente, mas ele falhará quando não tiver permissões para fazê-lo.

Também deve tomar cuidado ao executar *Comandos de gerência*, pois eles devem ser executados sob o mesmo utilizador que o Weblate em si está a ser executado, caso contrário, permissões em alguns ficheiros podem estar erradas.

In the Docker container, all files in the `/app/data` volume have to be owned by the `weblate` user inside the container (UID 1000).

Veja também:

Servir ficheiros estáticos

2.1.5 Configuração de banco de dados para o Weblate

Recomenda-se a executar o Weblate com um servidor de banco de dados PostgreSQL.

Veja também:

Usar um poderoso mecanismo de banco de dados, Databases, *Migrando de outros bancos de dados para o PostgreSQL*

PostgreSQL

PostgreSQL é geralmente a melhor escolha para sites baseados em Django. É o banco de dados de referência usado para implementar a camada de banco de dados Django.

Nota: O Weblate usa a extensão trigram que deve ser instalada separadamente em alguns casos. Procure por `postgresql-contrib` ou um pacote com nome similar.

Veja também:

PostgreSQL notes

Criar um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate num banco de dados separado e separar a conta do utilizador:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Dica: Se não quiser fazer do utilizador do Weblate um superutilizador no PostgreSQL, pode omiti-lo. Nesse caso, terá que executar algumas das etapas de migração manualmente como um superutilizador do PostgreSQL no esquema Weblate usará:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
```

Configurar Weblate para usar PostgreSQL

O trecho `settings.py` para PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

A migração do banco de dados executa ALTER ROLE na função de banco de dados usada pelo Weblate. Na maioria dos casos, o nome da função corresponde ao nome de utilizador. Em configurações mais complexas, o nome da função é diferente do nome de utilizador e obterá um erro sobre a função não existente durante a migração do banco de dados (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Isso é conhecido por acontecer com o Azure Database para PostgreSQL, mas não está limitado a este ambiente. Defina `ALTER_ROLE` para alterar o nome da função que o Weblate deve alterar durante a migração do banco de dados.

MySQL e MariaDB

Dica: Alguns recursos do Weblate terão melhor desempenho com *PostgreSQL*. Isso inclui a memória de pesquisa e tradução, que ambos utilizam recursos de texto completo no banco de dados e a implementação do PostgreSQL é superior.

Weblate também pode ser usado com MySQL ou MariaDB, consulte MySQL notes e MariaDB notes para ressalvas ao uso do Django com elas. or causa das limitações, é recomendado usar *PostgreSQL* para novas instalações.

O Weblate requer MySQL pelo menos 5.7.8 ou MariaDB pelo menos 10.2.7.

A configuração seguinte é recomendada para Weblate:

- Use o conjunto de caracteres `utf8mb4` para permitir a representação de planos Unicode mais altos (por exemplo, emojis).
- Configure o servidor com `innodb_large_prefix` para permitir índices mais longos em campos de texto.
- Defina o nível de isolamento para `READ COMMITTED`.
- O modo SQL deve ser definido como `STRICT_TRANS_TABLES`.

MySQL 8.x, MariaDB 10.5.x ou mais recente têm uma configuração padrão razoável, de modo que nenhum ajuste de servidor deve ser necessário e tudo o que é necessário pode ser configurado no lado do cliente.

Abaixo está um exemplo `/etc/my.cnf.d/server.cnf` para um servidor com 8 GB de RAM. Essas configurações devem ser suficientes para a maioria das instalações. MySQL e MariaDB têm ajustáveis que irão aumentar o desempenho do seu servidor que são considerados desnecessários, a menos que esteja a planejar ter uma grande quantidade de utilizadores simultâneos a acessar o sistema. Consulte a documentação de vários fornecedores sobre esses detalhes.

É absolutamente crítico para reduzir problemas durante a instalação que a configuração `innodb_file_per_table` esteja configurada corretamente e o MySQL/MariaDB reiniciado antes de iniciar a instalação do Weblate.

```
[mysqld]
character-set-server = utf8mb4
character-set-client = utf8mb4
collation-server = utf8mb4_unicode_ci

datadir=/var/lib/mysql

log-error=/var/log/mariadb/mariadb.log

innodb_large_prefix=1
innodb_file_format=Barracuda
innodb_file_per_table=1
innodb_buffer_pool_size=2G
sql_mode=STRICT_TRANS_TABLES
```

Dica: No caso de estar a receber o erro `#1071 - Specified key was too long; max key length is 767 bytes`, atualize a sua configuração para incluir as configurações `innodb` acima e reinicie a sua instalação.

Dica: No caso de estar a receber o erro `#2006 - MySQL server has gone away`, configurar `CONN_MAX_AGE` pode ajudar.

Configurando Weblate para usar MySQL/MariaDB

O trecho de `settings.py` para MySQL e MariaDB:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "127.0.0.1",
        # Set to empty string for default
        "PORT": "3306",
        # In case you wish to use additional
        # connection options
        "OPTIONS": {},
    }
}
```

Também deve criar a conta de utilizador `weblate` no MySQL ou MariaDB antes de iniciar a instalação. Use os comandos abaixo para fazer isso:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Outras configurações

Configuração de e-mail de saída

O Weblate envia e-mails em várias ocasiões - para a ativação de contas e sobre várias notificações configuradas pelos utilizadores. Para isso, precisa de acesso a um servidor de SMTP.

A configuração do servidor de e-mail é configurada a usar essas configurações: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_SSL`, `EMAIL_USE_TLS`, `EMAIL_HOST_USER` e `EMAIL_PORT`. Os nomes deles são bastante autoexplicativos, mas pode encontrar mais informações na documentação do Django.

Dica: Caso tenha erro sobre autenticação não suportada (por exemplo, `SMTP AUTH extension not supported by server`), é provavelmente causada pelo uso de conexão insegura e o servidor se recusa a autenticar dessa forma. Tente ativar `EMAIL_USE_TLS` nesse caso.

Veja também:

Não a receber e-mails do Weblate, *Configurando e-mail de saída em contentor Docker*

Executar por trás de um proxy reverso

Vários recursos no Weblate dependem de ser capaz de obter o endereço IP do cliente. Isso inclui *Limitação de taxa*, *Proteção contra spam* ou *Registo de auditoria*.

Na configuração predefinida, o Weblate analisa o endereço IP de `REMOTE_ADDR` que é definido pelo manipulador WSGI.

Se estiver a usar um proxy reverso, este campo provavelmente conterá o seu endereço. Precisa configurar o Weblate para confiar em cabeçalhos HTTP adicionais e analisar o endereço IP destes. Isso não pode ser ativado por predefinição, porque permitiria a falsificação de endereços IP para instalações que não usam um proxy reverso. Ativar *`IP_BEHIND_REVERSE_PROXY`* pode ser suficiente para as configurações mais usuais, mas podia precisar de ajustar *`IP_PROXY_HEADER`* e *`IP_PROXY_OFFSET`* também.

Outra coisa a tratar é do cabeçalho Host. Deve combinar com o que estiver configurado como **:set:`SITE_DOMAIN`**. Uma configuração adicional pode ser necessária no seu proxy reverso (por exemplo, use `ProxyPreserveHost On` para o Apache ou `proxy_set_header Host $host;` no nginx).

Veja também:

Proteção contra spam, *Limitação de taxa*, *Registo de auditoria*, *`IP_BEHIND_REVERSE_PROXY`*, *`IP_PROXY_HEADER`*, *`IP_PROXY_OFFSET`*, `SECURE_PROXY_SSL_HEADER`

Proxy HTTP

O Weblate executa comandos VCS e esses que aceitam a configuração proxy do ambiente. A abordagem recomendada é definir configurações de proxy em `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Veja também:

Variáveis de ambiente de proxy

2.1.7 Ajustar a configuração

Veja também:

Amostra de configuração

Copie `weblate/settings_example.py` para `weblate/settings.py` e ajuste-o para corresponder à configuração. Provavelmente irá ajustar as opções a seguir: `ADMINS`

> Lista de administradores de sites para receber notificações quando algo dá errado, por exemplo, notificações em mesclagens fracassadas ou erros de Django.
>
> **Veja também:**
>
> `ADMINS`, *Configurar administradores corretamente*

`ALLOWED_HOSTS`

> Precisa definir isso para listar os hosts que o seu site deve servir. Por exemplo:
>
> ```
> ALLOWED_HOSTS = ["demo.weblate.org"]
> ```
>
> Alternativamente, pode incluir curinga:
>
> ```
> ALLOWED_HOSTS = ["*"]
> ```

Veja também:

ALLOWED_HOSTS, *WEBLATE_ALLOWED_HOSTS*, *Configuração de hosts permitidos*

SESSION_ENGINE

Configure como as suas sessões serão armazenadas. Caso mantenha o mecanismo de backend do banco de dados predefinido, deve agendar: **weblate clearsessions** para remover dados de sessão obsoletos do banco de dados.

Se estiver a usar o Redis como cache (veja *Ativar o cache*) é recomendado também usá-lo para sessões:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Veja também:

Configuring the session engine, SESSION_ENGINE

DATABASES

Conetividade ao servidor de banco de dados, verifique a documentação do Django para obter mais detalhes.

Veja também:

Configuração de banco de dados para o Weblate, DATABASES, Databases

DEBUG

Desative isto para qualquer servidor de produção. Com o modo de depuração ativado, o Django mostrará backtraces em caso de erro aos utilizadores, quando desativá-lo, erros serão enviados por e-mail para ADMINS (veja acima).

O modo de depuração também desacelera o Weblate, já que o Django armazena muito mais informações internamente neste caso.

Veja também:

DEBUG, *Desativar o modo de depuração*

DEFAULT_FROM_EMAIL

Endereço de remetente de e-mail para e-mail de saída, por exemplo, e-mails de registo.

Veja também:

DEFAULT_FROM_EMAIL

SECRET_KEY

Chave usada por Django para assinar informações em cookies, consulte *Chave secreta do Django* para obter mais informações.

Veja também:

SECRET_KEY

SERVER_EMAIL

E-mail usado como endereço de remetente para envio de e-mails ao administrador, por exemplo, notificações em mesclagens falhadas.

Veja também:

SERVER_EMAIL

2.1.8 Preencher o banco de dados

Depois que a sua configuração estiver pronta, pode executar `weblate migrate` para criar a estrutura do banco de dados. Agora deve ser capaz de criar projetos de tradução a usar a interface administrativa.

Caso quere executar uma instalação não interativamente, pode usar `weblate migrate --noinput` e depois criar um utilizador administrativo pelo comando `createadmin`.

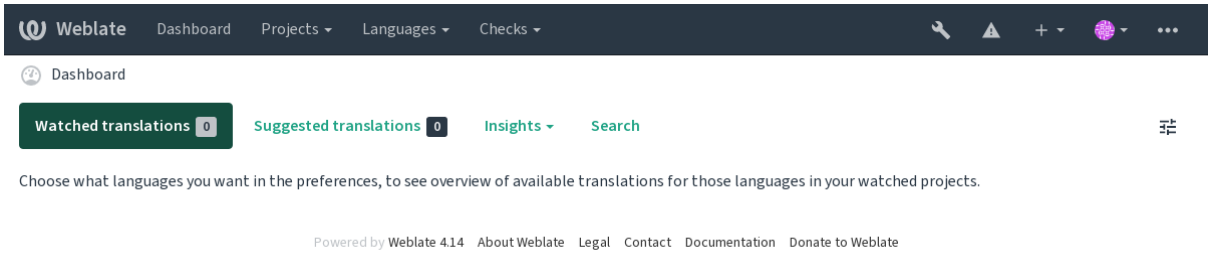
Uma vez feito, também deve verificar o *Relatório de desempenho* na interface administrativa, o que lhe dará dicas de configuração potencial não ideal no seu site.

Veja também:

Configuração, *Lista de privilégios e funções internas*

2.1.9 Configuração de produção

Para uma configuração de produção, deve realizar ajustes descritos nas seções a seguir. As configurações mais críticas acionarão um aviso, que é indicado por um ponto de exclamação na barra superior se esitver conectado como um superutilizador:



Também é recomendado inspecionar verificações desencadeadas por Django (embora possa não precisar corrigir todas):

```
weblate check --deploy
```

Também pode revisar a mesma lista de verificação na *Interface de gestão*.

Veja também:

Deployment checklist

Desativar o modo de depuração

Desative o modo de depuração do Django (`DEBUG`) com:

```
DEBUG = False
```

Com o modo de depuração ativado, o Django armazena todas as consultas executadas e mostra aos utilizadores backtraces de erros, o que não é desejado numa configuração de produção.

Veja também:

Ajustar a configuração

Configurar administradores corretamente

Defina os endereços de administração corretos à configuração *ADMINS* para definir quem receberá e-mails caso algo dê errado no servidor, por exemplo:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Veja também:

Ajustar a configuração

Definir domínio correto do site

Ajuste o nome e o domínio do site na interface administrativa, caso contrário, ligações no RSS ou e-mails de registo não funcionarão. Isto é configurado usando *SITE_DOMAIN* que deve conter o nome de domínio do site.

Alterado na versão 4.2: Antes da versão 4.2, a estrutura de sites do Django era usada em vez disso, consulte The "sites" framework.

Veja também:

Configuração de hosts permitidos, *Configurar HTTPS corretamente* *SITE_DOMAIN*, *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

Configurar HTTPS corretamente

É fortemente recomendado executar Weblate a com o protocolo criptografado HTTPS. Depois de ativá-lo, deve definir *ENABLE_HTTPS* nas configurações:

```
ENABLE_HTTPS = True
```

Dica: Pode também configurar o HSTS, consulte SSL/HTTPS para obter mais detalhes.

Veja também:

ENABLE_HTTPS, *Configuração de hosts permitidos*, *Definir domínio correto do site*

Definir SECURE_HSTS_SECONDS corretamente

Se o seu site for servido sobre SSL, deve considerar definir um valor para **:django:`SECURE_HSTS_SECONDS`** no `settings.py` para ativar HTTP Strict Transport Security. Por padrão, ele está definido para 0 como mostrado abaixo.

```
SECURE_HSTS_SECONDS = 0
```

Se for definido como um valor inteiro não-zero, o cabeçalho `django.middleware.security.SecurityMiddleware` define o cabeçalho HTTP Strict Transport Security em todas as respostas que ainda não o possuem.

Aviso: Definir isto incorretamente pode quebrar irreversivelmente (por algum tempo) o seu site. Leia primeiro a documentação HTTP Strict Transport Security.

Usar um poderoso mecanismo de banco de dados

- Por favor, use PostgreSQL para um ambiente de produção, consulte *Configuração de banco de dados para o Weblate* para obter mais informações.
- Use um local adjacente para executar o servidor de banco de dados, caso contrário, o desempenho ou confiabilidade da rede podem arruinar a sua experiência com o Weblate.
- Verifique o desempenho do servidor de banco de dados ou ajuste a sua configuração, por exemplo, usando PGTune.

Veja também:

Configuração de banco de dados para o Weblate, *Migrando de outros bancos de dados para o PostgreSQL*, *Ajustar a configuração*, Databases

Ativar o cache

Se for possível, use Redis do Django e ajuste a variável de configuração `CACHES`, por exemplo:

```
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/0",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # 'LOCATION': 'unix:///var/run/redis/redis.sock?db=0',
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    }
}
```

Dica: Caso altere as configurações do Redis para o cache, pode ser necessário ajustá-las também para o Celery, consulte *Tarefas de fundo a usar o Celery*.

Veja também:

Cache de avatares, Django's cache framework

Cache de avatares

Além do cache de Django, Weblate realiza cache de avatares. Recomenda-se usar um cache separado, baseado em ficheiros para este fim:

```
CACHES = {
    "default": {
        # Default caching backend setup, see above
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "unix:///var/run/redis/redis.sock?db=0",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
```

(continues on next page)

(continuação da página anterior)

```
        "TIMEOUT": 604800,
        "OPTIONS": {
            "MAX_ENTRIES": 1000,
        },
    },
}
```

Veja também:

ENABLE_AVATARS, *AVATAR_URL_PREFIX*, *Avatares*, *Ativar o cache*, Django's cache framework

Configurar envio de e-mail

O Weblate precisa enviar e-mails em várias ocasiões e esses e-mails devem ter um endereço de remetente correto, por favor, configure **:configuração:`SERVER_EMAIL`** e *DEFAULT_FROM_EMAIL* para combinar com o seu ambiente, por exemplo:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Nota: Para desativar o envio de e-mails pelo Weblate, defina `EMAIL_BACKEND` a `django.core.mail.backends.dummy.EmailBackend`.

Isso desativará *toda* a entrega de e-mail, incluindo e-mails de registo ou redefinição de palavra-passe.

Veja também:

Ajustar a configuração, *Configuração de e-mail de saída*, `EMAIL_BACKEND`, `DEFAULT_FROM_EMAIL`, `SERVER_EMAIL`

Configuração de hosts permitidos

Django requer *ALLOWED_HOSTS* para manter uma lista de nomes de domínio que o seu site pode servir, deixá-lo vazio bloqueará todas solicitações.

Caso isso não esteja configurado para corresponder ao seu servidor HTTP, terá erros como `Invalid HTTP_HOST header: '1.1.1.1'. Pode ter que adicionar '1.1.1.1' ao ALLOWED_HOSTS.`

Dica: No contentor Docker, isso está disponível como *WEBLATE_ALLOWED_HOSTS*.

Veja também:

ALLOWED_HOSTS, *WEBLATE_ALLOWED_HOSTS*, *Definir domínio correto do site*

Chave secreta do Django

A configuração *SECRET_KEY* é usada pelo Django para assinar cookies e você deve realmente gerar o seu próprio valor em vez de usar o da configuração do exemplo.

Pode gerar uma nova chave por **`weblate-generate-secret-key`**, que vem com o Weblate.

Veja também:

SECRET_KEY

Executar tarefas de manutenção

Para um desempenho ideal, é uma boa ideia executar algumas tarefas de manutenção em segundo plano. Isso agora é feito automaticamente por *Tarefas de fundo a usar o Celery* e abrange as seguintes tarefas:

- Verificação de saúde da configuração (de hora em hora).
- Realização de commits de alterações pendentes (de hora em hora), consulte *Commits adiados* e `commit_pending`.
- Atualização de alertas de componentes (dialy).
- Atualização dos ramos remotos (nightly), consulte `AUTO_UPDATE`.
- Backup de memória de tradução para JSON (diariamente), consulte `dump_memory`.
- Tarefas de manutenção de texto completo e banco de dados (tarefas diárias e semanais), consulte `cleanup-trans`.

Alterado na versão 3.2: Desde a versão 3.2, a maneira predefinida de executar essas tarefas é usar o Celery e o Weblate já vem com a configuração adequada, consulte *Tarefas de fundo a usar o Celery*.

Codificação e localidades do sistema

As localidades do sistema devem ser configuradas para UTF-8. Na maioria das distribuições Linux, esta é a configuração predefinida. Se não é o caso no seu sistema, altere as localidades para a variante UTF-8.

Por exemplo, a editar `/etc/default/locale` e a definir lá `LANG="C.UTF-8"`.

Em alguns casos, os serviços individuais têm configuração separada para locais. Isso varia entre a distribuição e os servidores da web, portanto, verifique a documentação dos pacotes do servidor da web para isso.

Apache no Ubuntu usa `/etc/apache2/envvars`:

```
export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'
```

Apache no CentOS usa `/etc/sysconfig/httpd` (ou `/opt/rh/httpd24/root/etc/sysconfig/httpd`):

```
LANG='en_US.UTF-8'
```

Usar uma autoridade certificadora personalizada

O Weblate verifica os certificados SSL durante as solicitações HTTP. Caso esteja a usar uma autoridade de certificação personalizada que não seja confiável em maços predefinidos, terá que adicionar o seu certificado como confiável.

A abordagem preferida é fazer isso no nível do sistema. Consulte a documentação da sua distro para mais detalhes (por exemplo, no Debian isso pode ser feito a por o certificado da AC em `/usr/local/share/ca-certificates/` e executando **`update-ca-certificates`**).

Uma vez feito isso, as ferramentas do sistema confiarão no certificado e isso inclui o Git.

Para código em Python, precisará configurar solicitações para usar o pacote de AC do sistema em vez do fornecido . Isso pode ser conseguido pondo os seguintes trechos em `settings.py` (o caminho é específico do Debian):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Comprimir os ativos do cliente

O Weblate vem com um monte de ficheiros JavaScript e CSS. Por razões de desempenho, é bom comprimi-los antes de enviar para um cliente. Na configuração predefinida isso é feito rapidamente ao custo de pouca sobrecarga. Em grandes instalações, recomenda-se ativar o modo de compressão offline. Isso precisa ser feito na configuração e a compressão tem que ser acionada em cada atualização do Weblate.

A mudança da configuração é simples ao ativar `django.conf.settings.COMPRESS_OFFLINE` e configuração `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (este último já está incluído na configuração do exemplo):

```
COMPRESS_OFFLINE = True
```

Em cada implantação precisa compactar os ficheiros para corresponder à versão atual:

```
weblate compress
```

Dica: A imagem oficial do Docker já tem este recurso ativado.

Veja também:

Common Deployment Scenarios, *Servir ficheiros estáticos*

2.1.10 Executar o servidor

Dica: No caso de não ter experiência com os serviços descritos abaixo, pode tentar seguir *Instalando a usar Docker*.

Precisará de vários serviços para executar o Weblate, a configuração recomendada consiste em:

- Servidor de banco de dados (consulte *Configuração de banco de dados para o Weblate*)
- Servidor de cache (consulte *Ativar o cache*)
- Servidor web frontend para ficheiros estáticos e terminação SSL (consulte *Servir ficheiros estáticos*)
- Servidor WSGI para conteúdo dinâmico (consulte *Configuração de amostra para NGINX e uWSGI*)
- Celery para executar tarefas em segundo plano (consulte *Tarefas de fundo a usar o Celery*)

Nota: Existem algumas dependências entre os serviços, por exemplo, o cache e o banco de dados devem estar em execução ao iniciar os processos de Celery ou uwsgi.

Na maioria dos casos, executará todos os serviços num único servidor (virtual), mas se a sua instalação estar muito carregada, pode dividir os serviços. A única limitação disso é que os servidores Celery e Wsgi precisam acessar `DATA_DIR`.

Nota: O processo de WSGI deve ser executado sob o mesmo utilizador que o processo do Celery, caso contrário, os ficheiros em `DATA_DIR` serão armazenados com propriedade mista, a levar a problemas de tempo de execução.

Veja também *Permissões do sistema de ficheiros* e *Tarefas de fundo a usar o Celery*.

Executar um servidor web

Executar o Weblate não é diferente de executar qualquer outro programa baseado em Django. Django é geralmente executado como uWSGI ou fcgi (consulte exemplos para diferentes servidores web abaixo).

Para fins de teste, pode usar o servidor web incorporado no Django:

```
weblate runserver
```

Aviso: NÃO USE ESTE SERVIDOR NUMA CONFIGURAÇÃO DE PRODUÇÃO. Não passou por auditorias de segurança ou testes de desempenho. Veja também a documentação de Django no `runserver`.

Dica: O servidor embutido do Django serve apenas ficheiros estáticos com *DEBUG* ativado, pois é destinado apenas ao desenvolvimento. Para uso em produção, consulte as configurações de wsgi em *Configuração de amostra para NGINX e uWSGI*, *Configuração de amostra para Apache*, *Configuração de amostra para Apache and Gunicorn* e *Servir ficheiros estáticos*.

Servir ficheiros estáticos

Alterado na versão 2.4: Antes da versão 2.4, o Weblate não usava a estrutura de ficheiros estáticos do Django corretamente e a configuração era mais complexa.

Django precisa coletar os ficheiros estáticos dele num único diretório. Para isso, execute `weblate collectstatic --noinput`. Isso copiará os ficheiros estáticos num diretório especificado pela configuração `STATIC_ROOT` (isso é a predefinição para um diretório `static` dentro de *DATA_DIR*).

Recomenda-se servir ficheiros estáticos diretamente do seu servidor web. Deve usá-los para os seguintes caminhos:

/static/
Serve ficheiros estáticos para Weblate e a interface de administração (definida por `STATIC_ROOT`).

/media/
Usado para o envio de mídia pelo utilizador (por exemplo, capturas de ecrã).

/favicon.ico
Deve ser reescrito para reescrever uma regra para servir `/static/favicon.ico`.

Veja também:

Configuração de amostra para NGINX e uWSGI, *Configuração de amostra para Apache*, *Configuração de amostra para Apache and Gunicorn*, *Comprimir os ativos do cliente*, How to deploy Django, How to deploy static files

Política de segurança de conteúdo

A configuração predefinida do Weblate ativa o middleware `weblate.middleware.SecurityMiddleware` que define cabeçalhos HTTP relacionados à segurança como `Content-Security-Policy` ou `X-XSS-Protection`. São configurados por predefinição para funcionar com o Weblate e a configuração dele, mas isso pode precisar de personalização no seu ambiente.

Veja também:

CSP_SCRIPT_SRC, *CSP_IMG_SRC*, *CSP_CONNECT_SRC*, *CSP_STYLE_SRC*, *CSP_FONT_SRC*

Configuração de amostra para NGINX e uWSGI

Para executar o servidor web de produção, use o wrapper wsgi instalado com Weblate (no caso de ambiente virtual é instalado como `~/weblate-env/lib/python3.9/site-packages/weblate/wsgi.py`). Também não se esqueça de definir o caminho de pesquisa Python para o seu virtualenv (por exemplo, usando *virtualenv = /home/user/weblate-env`* no uWSGI).

A configuração a seguir executa o Weblate como uWSGI sob o servidor web NGINX.

A configuração para NGINX (também disponível como `weblate/examples/weblate.nginx.conf`):

```
#
# nginx configuration for Weblate
#
# You will want to change:
#
# - server_name
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
server {
    listen 80;
    server_name weblate;
    # Not used
    root /var/www/html;

    location ~ ^/favicon.ico$ {
        # DATA_DIR/static/favicon.ico
        alias /home/weblate/data/static/favicon.ico;
        expires 30d;
    }

    location /static/ {
        # DATA_DIR/static/
        alias /home/weblate/data/static/;
        expires 30d;
    }

    location /media/ {
        # DATA_DIR/media/
        alias /home/weblate/data/media/;
        expires 30d;
    }

    location / {
        include uwsgi_params;
        # Needed for long running operations in admin interface
        uwsgi_read_timeout 3600;
        # Adjust based to uwsgi configuration:
        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`):

```
#
# uWSGI configuration for Weblate
#
# You will want to change:
#
```

(continues on next page)

(continuação da página anterior)

```
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
[uwsgi]
plugins       = python3
master        = true
protocol      = uwsgi
socket        = 127.0.0.1:8080
wsgi-file     = /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/wsgi.
↪py

# Add path to Weblate checkout if you did not install
# Weblate by pip
# python-path   = /path/to/weblate

# In case you're using virtualenv uncomment this:
virtualenv = /home/weblate/weblate-env

# Needed for OAuth/OpenID
buffer-size   = 8192

# Reload when consuming too much of memory
reload-on-rss = 250

# Increase number of workers for heavily loaded sites
workers       = 8

# Enable threads for Sentry error submission
enable-threads = true

# Child processes do not need file descriptors
close-on-exec = true

# Avoid default 0000 umask
umask = 0022

# Run as weblate user
uid = weblate
gid = weblate

# Enable harakiri mode (kill requests after some time)
# harakiri = 3600
# harakiri-verbose = true

# Enable uWSGI stats server
# stats = :1717
# stats-http = true

# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

Veja também:

How to use Django with uWSGI

Configuração de amostra para Apache

Recomenda-se o uso de prefork de MPM ao usar WSGI com Weblate.

A configuração a seguir executa o Weblate como WSGI. Precisa ter ativado `mod_wsgi` (disponível como `weblate/examples/apache.conf`):

```
#
# VirtualHost for Weblate
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate␣
↪request-timeout=600
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias / /home/weblate/weblate-env/lib/python3.9/site-packages/
↪weblate/wsgi.py process-group=weblate
    WSGIPassAuthorization On

    <Directory /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/>
        <Files wsgi.py>
        Require all granted
        </Files>
    </Directory>

</VirtualHost>
```

Nota: Weblate precisa do Python 3, então, por favor se certifique que está a executar a variante do Python 3 do modwsgi. Usualmente, está disponível como um pacote separado, por exemplo `libapache2-mod-wsgi-py3`.

Veja também:

Codificação e localidades do sistema, How to use Django with Apache and mod_wsgi

Configuração de amostra para Apache and Gunicorn

A configuração seguinte executa o Weblate em Gunicorn and Apache 2.4 (disponível como `weblate/examples/apache.gunicorn.conf`):

```
#
# VirtualHost for Weblate using gunicorn on localhost:8000
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:443>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/https_cert.cert
    SSLCertificateKeyFile /etc/apache2/ssl/https_key.pem
    SSLProxyEngine On

    ProxyPass /favicon.ico !
    ProxyPass /static/ !
    ProxyPass /media/ !

    ProxyPass / http://localhost:8000/
    ProxyPassReverse / http://localhost:8000/
    ProxyPreserveHost On
</VirtualHost>
```

Veja também:

How to use Django with Gunicorn

A executar o Weblate sob o caminho

Novo na versão 1.3.

Recomenda-se o uso de prefork de MPM ao usar WSGI com Weblate.

Uma configuração do Apache de amostra para servir o Weblate sob `/weblate`. Novamente, a usar `mod_wsgi` (também disponível como `weblate/examples/apache-path.conf`):

```
#
# VirtualHost for Weblate, running under /weblate path
#
# You will want to change:
#
# - ServerAdmin and ServerName
# - change /home/weblate/weblate-env to location where Weblate virtualenv is placed
# - change /home/weblate/data to match your DATA_DIR
# - change python3.9 to match your Python version
# - change weblate user to match your Weblate user
#
<VirtualHost *:80>
    ServerAdmin admin@weblate.example.org
    ServerName weblate.example.org

    # DATA_DIR/static/favicon.ico
    Alias /weblate/favicon.ico /home/weblate/data/static/favicon.ico

    # DATA_DIR/static/
    Alias /weblate/static/ /home/weblate/data/static/
    <Directory /home/weblate/data/static/>
        Require all granted
    </Directory>

    # DATA_DIR/media/
    Alias /weblate/media/ /home/weblate/data/media/
    <Directory /home/weblate/data/media/>
        Require all granted
    </Directory>

    # Path to your Weblate virtualenv
    WSGIDaemonProcess weblate python-home=/home/weblate/weblate-env user=weblate␣
↪request-timeout=600
    WSGIProcessGroup weblate
    WSGIApplicationGroup %{GLOBAL}

    WSGIScriptAlias /weblate /home/weblate/weblate-env/lib/python3.9/site-packages/
↪weblate/wsgi.py process-group=weblate
    WSGIPassAuthorization On

    <Directory /home/weblate/weblate-env/lib/python3.9/site-packages/weblate/>
        <Files wsgi.py>
        Require all granted
        </Files>
    </Directory>

</VirtualHost>
```

Adicionalmente, irá ter de ajustar o `weblate/settings.py`:

```
URL_PREFIX = "/weblate"
```

2.1.11 Tarefas de fundo a usar o Celery

Novo na versão 3.2.

Weblate usa Celery para executar tarefas regulares e em segundo plano. Deve executar o serviço Celery que irá executá-los. Por exemplo, é responsável por lidar com as seguintes operações (esta lista não está completa):

- Receber webhooks de serviços externos (veja *Hooks de notificação*).
- Executar tarefas de manutenção regulares como backups, limpezas, extensões diárias ou atualizações (veja *Fazer backup e mover o Weblate*, `BACKGROUND_TASKS`, *Extensões*).
- Executar *Tradução automática*.
- Enviar notificações de resumo.
- Descarregar operações caras do processo wsgi.
- Fazer commits de alterações pendentes (veja *Commits adiados*).

Uma configuração típica a usar redis como backend se parece com isso:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Veja também:

Configuração do broker Redis no Celery

Também deve iniciar o worker do Celery para processar as tarefas e iniciar tarefas programadas, isso pode ser feito diretamente na linha de comando (o que é mais útil ao depurar ou desenvolver):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Nota: O processo de Celery deve ser executado sob o mesmo utilizador que o Weblate e o processo do WSGI, caso contrário, os ficheiros em `DATA_DIR` serão armazenados com propriedade mista, a levar a problemas de tempo de execução.

Veja também *Permissões do sistema de ficheiros* e *Executar o servidor*.

Execução de tarefas de Celery no wsgi a usar o modo ansioso

Nota: Isso terá um impacto severo no desempenho da interface da web e interromperá os recursos a depender do acionamento normal (por exemplo, fazer commit de alterações pendentes, notificações de resumo ou backups).

Para o desenvolvimento, pode usar uma configuração ansiosa, que processa todas as tarefas no local:

```
CELERY_TASK_ALWAYS_EAGER = True
CELERY_BROKER_URL = "memory://"
CELERY_TASK_EAGER_PROPAGATES = True
```

Executando Celery como serviço do sistema

Provavelmente vai querer executar o Celery como um daemon e que é coberto por Daemonization. Para as configurações Linux mais comuns a usar systemd, pode usar os ficheiros de exemplo disponibilizados na pasta `examples` listado abaixo.

Unidade do systemd a ser posta como `/etc/systemd/system/celery-weblate.service`:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target

[Service]
Type=forking
User=weblate
Group=weblate
EnvironmentFile=/etc/default/celery-weblate
WorkingDirectory=/home/weblate
RuntimeDirectory=celery
RuntimeDirectoryPreserve=restart
LogsDirectory=celery
ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES} \
  --pidfile=${CELERYD_PID_FILE}'
ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'

[Install]
WantedBy=multi-user.target
```

Configuração do ambiente a ser posta como `/etc/default/celery-weblate`:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-multiplier:celery=4 \
    --queues:notify=notify --prefetch-multiplier:notify=10 \
    --queues:memory=memory --prefetch-multiplier:memory=10 \
    --queues:translate=translate --prefetch-multiplier:translate=4 \
    --concurrency:backup=1 --queues:backup=backup  --prefetch-multiplier:backup=2"

# Logging configuration
# - %n will be replaced with the first part of the nodename.
# - %I will be replaced with the current child process index
#   and is important when using the prefork pool to avoid race conditions.
CELERYD_PID_FILE="/run/celery/weblate-%n.pid"
CELERYD_LOG_FILE="/var/log/celery/weblate-%n%I.log"
CELERYD_LOG_LEVEL="INFO"
```

Configuração adicional para alternar os logs do Celery a usar **`logrotate`** a ser posta como `/etc/logrotate.d/celery`:

```
/var/log/celery/*.log {
        weekly
        missingok
        rotate 12
        compress
        notifempty
}
```

Tarefas periódicas a usar a batida do Celery

O Weblate vem com configuração embutida para tarefas programadas. No entanto, pode definir tarefas adicionais em `settings.py`. Por exemplo, veja *Commits adiados*.

As tarefas deveriam ser executadas por daemons de batidas do Celery. Caso não esteja a funcionar corretamente, pode não estar a funcionar ou o seu banco de dados foi corrompido. Verifique os registos de inicialização do Celery nesse caso para descobrir a causa raiz.

Monitorar o estado do Celery

Pode encontrar o comprimento atual das filas de tarefas do Celery na *Interface de gestão* ou pode usar `celery_queues` na linha de comando. Caso a fila fique muito longa, também terá erro de configuração na interface administrativa.

> **Aviso:** Os erros do Celery são por padrão apenas conectados ao log do Celery e não são visíveis ao utilizador. Caso queira ter uma visão geral sobre tais falhas, recomenda-se ajustar a configuração para ir *Coletando relatórios de erros*.

Veja também:

Monitorando o Weblate, *Como posso verificar se a minha Weblate está configurada corretamente?*, Configuration and defaults, Workers Guide, Daemonization, Monitoring and Management Guide, `celery_queues`

2.1.12 Monitorando o Weblate

O Weblate fornece a URL `/healthz/` a ser usada em verificações de saúde simples, por exemplo, usando Kubernetes. O contentor Docker tem verificação de saúde embutida usando esta URL.

Para monitorar as métricas do Weblate, pode usar o ponto final `GET /api/metrics/` da API.

Veja também:

Como posso verificar se a minha Weblate está configurada corretamente?, *Monitorar o estado do Celery*, Weblate plugin for Munin

2.1.13 Coletando relatórios de erros

Weblate, como qualquer outro software, pode falhar. Para coletar estados de falha úteis, recomendamos usar serviços de terceiros para coletar tais informações. Isso é especialmente útil no caso de falhas nas tarefas do Celery, que de outra forma só relatariam erro nos logs e não será notificado sobre eles. O Weblate tem suporte para os seguintes serviços:

Sentry

O Weblate possui suporte embutido para Sentry. Para usá-lo, é suficiente definir *SENTRY_DSN* no `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

O Weblate tem suporte embutido para Rollbar. Para usá-lo, basta seguir instruções para o notificador de Rollbar para Python.

Em suma, precisa ajustar `settings.py`:

```
# Add rollbar as last middleware:
MIDDLEWARE = [
    # … other middleware classes …
    "rollbar.contrib.django.middleware.RollbarNotifierMiddleware",
]

# Configure client access
ROLLBAR = {
    "access_token": "POST_SERVER_ITEM_ACCESS_TOKEN",
    "client_token": "POST_CLIENT_ITEM_ACCESS_TOKEN",
    "environment": "development" if DEBUG else "production",
    "branch": "main",
    "root": "/absolute/path/to/code/root",
}
```

Todo o resto é integrado automaticamente, agora coletará erros do lado do servidor e do cliente.

2.1.14 Migrando Weblate para outro servidor

Migrar o Weblate para outro servidor deve ser muito fácil, porém armazena dados em poucos locais que deve migrar cuidadosamente. A melhor abordagem é parar o Weblate para a migração.

Migrando banco de dados

Dependendo do backend do seu banco de dados, pode ter várias opções para migrar o banco de dados. O mais simples é despejar o banco de dados num servidor e importá-lo no novo. Alternativamente, pode usar a replicação no caso do seu banco de dados ter suporte a isso.

A melhor abordagem é usar ferramentas nativas de banco de dados, pois geralmente são as mais eficazes (por exemplo, **mysqldump** ou **pg_dump**). Se quiser migrar entre diferentes bancos de dados, a única opção pode ser usar a gestão do Django para despejar e importar o banco de dados:

```
# Export current data
weblate dumpdata > /tmp/weblate.dump
# Import dump
weblate loaddata /tmp/weblate.dump
```

Migrando repositórios VCS

Os repositórios VCS armazenados em `DATA_DIR` também precisam ser migrados. Pode simplesmente copiá-los ou usar **`rsync`** para fazer a migração de forma mais eficaz.

Outras notas

Não se esqueça de mover outros serviços que o Weblate pode ter usado como Redis, tarefas do cron ou backends de autenticação personalizada.

2.2 Implantações de Weblate

O Weblate pode ser facilmente instalado na sua nuvem. Encontre um guia detalhado para sua plataforma:

- *Instalando a usar Docker*
- *Instalando no OpenShift*
- *Instalando no Kubernetes*

2.2.1 Implantações de terceiros para Weblate

Nota: As seguintes implantações não são desenvolvidas ou suportadas pela equipa Weblate. Partes da configuração podem variar do que está descrito nesta documentação.

Pilha Weblate para Bitnami

Bitnami fornece uma pilha Weblate para muitas plataformas em <https://bitnami.com/stack/weblate>. A configuração será ajustada durante a instalação, consulte <https://bitnami.com/stack/weblate/README.txt> para mais documentação.

Pacote de Weblate para Cloudron

Cloudron é uma plataforma para auto-hospedagem de aplicações da web. Weblate instalado com Cloudron será automaticamente atualizado. O pacote é mantido pela equipa Cloudron no seu *repositório de pacotes Weblate <https://git.cloudron.io/cloudron/weblate-app>* _.

Cloudron install

Weblate no YunoHost

O projeto de hospedagem própria YunoHost fornece um pacote para Weblate. Uma vez que tenha a sua instalação YunoHost, pode instalar o Weblate como qualquer outra aplicação. Ele fornecerá uma pilha de trabalho completo com backup e restauração, mas ainda pode ter que editar seu ficheiro de configurações para usos específicos.

Pode usar a sua interface de administração ou este botão (vai levá-lo ao seu servidor):

Install with YUNOHOST

Também é possível usar a interface da linha de comando:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Atualizando o Weblate

2.3.1 Atualizações de imagem Docker

A imagem Docker oficial (veja *Instalando a usar Docker*) contém todas as etapas de atualização integradas do Weblate. Normalmente não há nenhuma etapa manual necessária além de puxar a versão mais recente.

Veja também:

Atualizando o contentor Docker

2.3.2 Instruções genéricas de atualização

Antes de atualizar, verifique os atuais *Requisitos de software*, pois eles podem ter mudado. Uma vez que todos os requisitos estejam instalados ou atualizados, ajuste o seu `settings.py` para corresponder às mudanças na configuração (consulte `settings_example.py` para os valores corretos).

Sempre verifique *Instruções específicas de versão* antes de atualizar. Caso esteja a pular algumas versões, siga as instruções para todas as versões que está a pular nessa atualização. Às vezes, é melhor atualizar para alguma versão intermediária para garantir uma migração tranquila. A atualização em várias versões deve funcionar, mas não é tão bem testada quanto as atualizações de versão única.

Nota: É recomendado fazer um backup completo do banco de dados antes da atualização para que possa reverter o banco de dados caso a atualização falhe, veja *Fazer backup e mover o Weblate*.

1. Pare os processos wsgi e Cellery. A atualização pode realizar alterações incompatíveis no banco de dados, por isso é sempre mais seguro evitar a execução de processos antigos durante a atualização.
2. Atualize o código do Weblate.

   Para instalações pip, isso pode ser alcançado da seguinte forma:

   ```
   pip install -U "Weblate[all]==version"
   ```

   Or, if you just want to get the latest released version:

   ```
   pip install -U "Weblate[all]"
   ```

   Se não quiser instalar todas as dependências opcionais, faça:

   ```
   pip install -U Weblate
   ```

   Com um checkout do Git, precisa obter o novo código-fonte e atualizar a sua instalação:

   ```
   cd weblate-src
   git pull
   # Update Weblate inside your virtualenv
   . ~/weblate-env/bin/pip install -e .
   # Install dependencies directly when not using virtualenv
   pip install --upgrade -r requirements.txt
   # Install optional dependencies directly when not using virtualenv
   pip install --upgrade -r requirements-optional.txt
   ```

3. A nova versão do Weblate pode ter novas *Dependências opcionais*. Verifique se elas cobrem os recursos que deseja.

4. Atualize o ficheiro de configuração, consulte o `settings_example.py` ou :ref:`version-specific-instructions`para os passos necessários.
5. Atualize estrutura do banco de dados:

```
weblate migrate --noinput
```

6. Colete ficheiros estáticos atualizados (veja *Executar o servidor* e *Servir ficheiros estáticos*):

```
weblate collectstatic --noinput --clear
```

7. Compactar ficheiros JavaScript e CSS (opcional, veja *Comprimir os ativos do cliente*):

```
weblate compress
```

8. Se está a usar uma versão do Git, também deve gerar novamente ficheiros locais toda vez que estiver atualizando. Pode fazer isso a invocar:

```
weblate compilemessages
```

9. Verifique que a sua configuração é sana (ver também *Configuração de produção*):

```
weblate check --deploy
```

10. Reinicie o worker do Celery (veja *Tarefas de fundo a usar o Celery*).

2.3.3 Instruções específicas de versão

Atualizar de 2.x

Se está a atualizar da versão 2.x, sempre atualize para a 3.0.1 antes e depois continue atualizando na série 3.x. Atualizações pulando esse passo não são suportadas e não funcionarão corretamente.

Veja também:

Atualizar de 2.20 para 3.0 na documentação do Weblate 3.0

Atualizar da 3.x

Se está a atualizar da versão 3.x, sempre atualize para a 4.0.4 ou 4.1.1 antes e depois continue a atualizar na série 4.x. Atualizações a pular esse passo não são suportadas e não funcionarão corretamente.

Veja também:

Atualizar de 3.11 para 4.0 na documentação do Weblate 4.0

Atualizar da 4.0 para 4.1

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Mudanças notáveis de configuração ou dependências:

- Há várias mudanças em `settings_example.py`, notadamente de middleware. Por favor, ajuste as suas configurações de acordo.
- Há novos formatos de fiheiros, pode incluí-los no caso de ter modificado o `WEBLATE_FORMATS`.
- Há novas verificações de qualidade, pode incluí-las no caso de ter modificado o `CHECK_LIST`.
- Há uma mudança na configuração `DEFAULT_THROTTLE_CLASSES` para permitir relatórios de limitação de taxa na API.

- Há alguns requisitos novos e atualizados.
- Há uma mudança em `django: INSTALLED_APPS`.
- The `MT_DEEPL_API_VERSION` setting has been removed in Version 4.7. The *DeepL* machine translation now uses the new `MT_DEEPL_API_URL` instead. You might need to adjust `MT_DEEPL_API_URL` to match your subscription.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.1 para 4.2

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Mudanças notáveis de configuração ou dependências:

- A atualização de versões 3.x não é mais suportada. Atualize para 4.0 ou 4.1 primeiro.
- Há alguns requisitos novos e atualizados.
- Há várias mudanças em `settings_example.py`, notadamente um novo middleware e a mudança na ordem dos aplicações.
- As chaves para formatos baseados em JSON não incluem mais um ponto inicial. As sequências são ajustadas durante a migração do banco de dados, mas os componentes externos podem precisar de ajustes no caso de depender de chaves nas exportações ou API.
- A configuração do Celery foi alterada para não usar mais a fila `memory`. Ajuste os seus scripts de inicialização e configuração de `CELERY_TASK_ROUTES`.
- O domínio Weblate está agora configurado nas configurações, veja *SITE_DOMAIN* (ou *WEBLATE_SITE_DOMAIN*). Terá que configurá-lo antes de executar o Weblate.
- Os campos de nome de utilizador e e-mail no banco de dados de utilizadores agora devem ser únicos e sem distinção de maiúsculo para minúsculo. Isso acidentalmente não foi reforçado com PostgreSQL.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.2 para 4.3

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Mudanças notáveis de configuração ou dependências:

- Há algumas mudanças nas verificações de qualidade, pode incluí-las no caso de ter modificado o *CHECK_LIST*.
- O atributo do idioma fonte foi movido do projeto para um componente que é exposto na API. Precisará atualizar *Cliente Weblate* caso esteja a usar.
- A migração do banco de dados para 4.3 pode demorar, a depender da quantidade de cadeias que está a traduzir (espere cerca de uma hora de tempo de migração por 100.000 cadeias fonte).
- Há uma mudança em `django: INSTALLED_APPS`.
- Há uma nova configuração *SESSION_COOKIE_AGE_AUTHENTICATED* que complementa `SESSION_COOKIE_AGE`.
- Caso esteja a usar **`hub`** ou **`lab`** para integrar com GitHub ou GitLab, precisará reconfigurar isto, veja *GITHUB_CREDENTIALS* e *GITLAB_CREDENTIALS*.

Alterado na versão 4.3.1:

- A configuração do Celery foi alterada para adicionar a fila de `memory`. Ajuste os seus scripts de inicialização e configuração de `CELERY_TASK_ROUTES`.

Alterado na versão 4.3.2:

- O método `post_update` de extensões agora leva o parâmetro adicional `skip_push`.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.3 para 4.4

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Mudanças notáveis de configuração ou dependências:

- Há uma mudança em `django: INSTALLED_APPS`, `weblate.configuration` deve ser adicionado lá.
- Django 3.1 agora é necessário.
- Caso esteja a usar MySQL ou MariaDB, as versões mínimas necessárias aumentaram, veja *MySQL e MariaDB*.

Alterado na versão 4.4.1:

- *Gettext monolíngue* now uses both `msgid` and `msgctxt` when present. This will change identification of translation strings in such files breaking links to Weblate extended data such as screenshots or review states. Please make sure you commit pending changes in such files prior upgrading and it is recommended to force loading of affected component using `loadpo`.
- Incrementada a versão mínima necessária do translate-toolkit para resolver vários problemas de formato de ficheiros.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.4 para 4.5

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Mudanças notáveis de configuração ou dependências:

- A migração pode levar um tempo considerável se tiver grandes glossários.
- Glossários agora são armazenados como componentes normais.
- A API do glossário foi removida. Use a API de tradução normal para acessar os glossários.
- Há uma mudança em `django: INSTALLED_APPS` - `weblate.metrics` deve ser adicionado.

Alterado na versão 4.5.1:

- Há uma nova dependência do módulo *pyahocorasick*.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.5 para 4.6

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Mudanças notáveis de configuração ou dependências:

- Há novos formatos de fiheiros, pode incluí-los no caso de ter modificado o `WEBLATE_FORMATS`.
- A API para a criação de componentes agora usa automaticamente *URLs internas do Weblate*, veja `POST /api/projects/(string:project)/components/`.
- Há uma mudança nas dependências e `django: PASSWORD_HASHERS` para preferir Argon2 para hashing de palavras-passe.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.6 para 4.7

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Mudanças notáveis de configuração ou dependências:

- Há várias mudanças em `settings_example.py`, notadamente mudanças de middleware (`django: MIDDLEWARE`). Ajuste as suas configurações de acordo.
- The *DeepL* machine translation now has a generic `MT_DEEPL_API_URL` setting to adapt to different subscription models more flexibly. The `MT_DEEPL_API_VERSION` setting is no longer used.
- Django 3.2 agora é necessário.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.7 para 4.8

Siga as *Instruções genéricas de atualização* para realizar a atualização.

Não há etapas de atualização adicionais necessárias nesta versão.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.8 para 4.9

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- Há uma mudança no armazenamento de métricas, a atualização pode levar muito tempo em sites maiores.

Veja também:

Instruções genéricas de atualização

Atualizar da 4.9 para 4.10

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- Há uma mudança nos grupos por projeto, a atualização pode levar muito tempo em sites com milhares de projetos.
- Django 4.0 fez algumas mudanças incompatíveis, veja Backwards incompatible changes in 4.0. Weblate ainda oferece suporte ao Django 3.2 por enquanto, no caso de alguma dessas ser problemática. Mudanças mais notáveis que podem afetar o Weblate:
  - Removido o suporte a PostgreSQL 9.6, o Django 4.0 oferece suporte para PostgreSQL 10 e superior.
  - O formato de `CSRF_TRUSTED_ORIGINS` foi alterado.
- O contentor do Docker agora usa Django 4.0, consulte as alterações acima.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.10 to 4.11

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- Weblate now requires Python 3.7 or newer.
- A implementação de *A gerir controle de acesso por projeto* foi alterada, removendo o prefixo do projeto dos nomes dos grupos. Isso afeta os utilizadores da API.
- Weblate agora usa o módulo `charset-normalizer` em vez de `chardet` para detecção de conjunto de caracteres.
- **Alterado no 4.11.1:** Há uma alteração na configuração `REST_FRAMEWORK` (remoção de um dos backends em `DEFAULT_AUTHENTICATION_CLASSES`).

Veja também:

Instruções genéricas de atualização

Upgrade from 4.11 to 4.12

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- There are no special steps required.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.12 to 4.13

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- Os *Definições de idioma* agora são atualizados automaticamente na atualização, use `UPDATE_LANGUAGES` para desativar isso.
- Handling of context and location has been changed for *Ficheiros RC do Windows*, *Ficheiro HTML*, *Formato IDML*, and *Ficheiros de texto* file formats. In most cases the context is now shown as location.
- The machine translation services are now configured using the user interface, settings from the configuration file will be imported during the database migration.

Veja também:

Instruções genéricas de atualização

Upgrade from 4.13 to 4.14

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- The Java formatting checks now match GNU gettext flags. The flags set in Weblate will be automatically migrated, but third-party scripts will need to use `java-printf-format` instead of `java-format` and `java-format` instead of `java-messageformat`.
- The *jellyfish* dependency has been replaced by *rapidfuzz*.

Veja também:

Instruções genéricas de atualização

2.3.4 Atualização de Python 2 para Python 3

O Weblate não oferece mais suporte a Python anterior a 3.6. Caso ainda esteja a executar uma versão mais antiga, primeiro migre ao Python 3 na versão existente e atualize mais tarde. Consulte Atualizar do Python 2 ao Python 3 na documentação do Weblate 3.11.1.

2.3.5 Migrando de outros bancos de dados para o PostgreSQL

Se estiver a executar o Weblate com outro banco de dados que não é o PostgreSQL, deve considerar a migração ao PostgreSQL, pois o Weblate tem melhor desempenho com ele. As etapas a seguir o orientarão na migração dos seus dados entre os bancos de dados. Lembre-se de parar os servidores web e Celery antes da migração, caso contrário, pode acabar com dados inconsistentes.

Criar um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate num banco de dados separado e separar a conta do utilizador:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Migrando a usar despejos do Django em JSON

A abordagem mais simples para migração é utilizar despejos do Django em JSON. Isso funciona bem para instalações menores. Em sites maiores pode usar o pgloader, veja *Migrar para PostgreSQL usa usargloader*.

1. Adicione PostgreSQL como conexão de banco de dados adicional ao `settings.py`:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
```

(continues on next page)

(continuação da página anterior)

```
        "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. Execute migrações e descarte todos os dados inseridos nas tabelas:

```
weblate migrate --database=postgresql
weblate sqlflush --database=postgresql | weblate dbshell --database=postgresql
```

3. Despeje o banco de dados legado e importe no PostgreSQL

```
weblate dumpdata --all --output weblate.json
weblate loaddata weblate.json --database=postgresql
```

4. Ajuste `django: DATABASES` para usar apenas o banco de dados PostgreSQL como padrão, remova a conexão legada.

O Weblate agora deve estar pronto para ser executado a partir do banco de dados PostgreSQL.

Migrar para PostgreSQL usa usargloader

O pgloader é uma ferramenta de migração genérica para migrar dados para PostgreSQL. Pode usá-lo para migrar o banco de dados Weblate.

1. Ajuste o seu `settings.py` para usar o PostgreSQL como banco de dados.
2. Migre o esquema no banco de dados PostgreSQL:

   ```
   weblate migrate
   weblate sqlflush | weblate dbshell
   ```

3. Execute o pgloader para transferir os dados. O seguinte script pode ser usado para migrar o banco de dados, mas pode aprender mais sobre pgloader para entender o que ele faz e ajustá-lo para corresponder à sua configuração:

```
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 Migrando de Pootle

Como o Weblate foi originalmente escrito como substituto do Pootle, é compatível com a migração de contas de utilizador do Pootle. Pode descartar os utilizadores do Pootle e importá-los a usar `importusers`.

2.4 Fazer backup e mover o Weblate

2.4.1 Project level backups

Novo na versão 4.14.

Aviso: Restoring backups is only supported when using PostgreSQL or MariaDB 10.5+ as a database.

The project backups all translation content from Weblate (project, components, translations, string comments, suggestions or checks). It is suitable for transferring a project to another Weblate instance.

You can perform a project backup in *Manage* ↓ *Backups*. The backup can be restored when creating a project (see *Adicionando projetos e componentes de tradução*).

The backups currently do not include access control information and history.

The comments and suggestions are backed up with an username of user who did create them. Upon import it is assigned to a matching user. If there is no user with such username, it is assigned to anonymous user.

The generated backups are kept on the server as configured by `PROJECT_BACKUP_KEEP_DAYS` and `PROJECT_BACKUP_KEEP_COUNT` (it defaults to keep at most 3 backups for 30 days).

2.4.2 Backup automatizado pelo BorgBackup

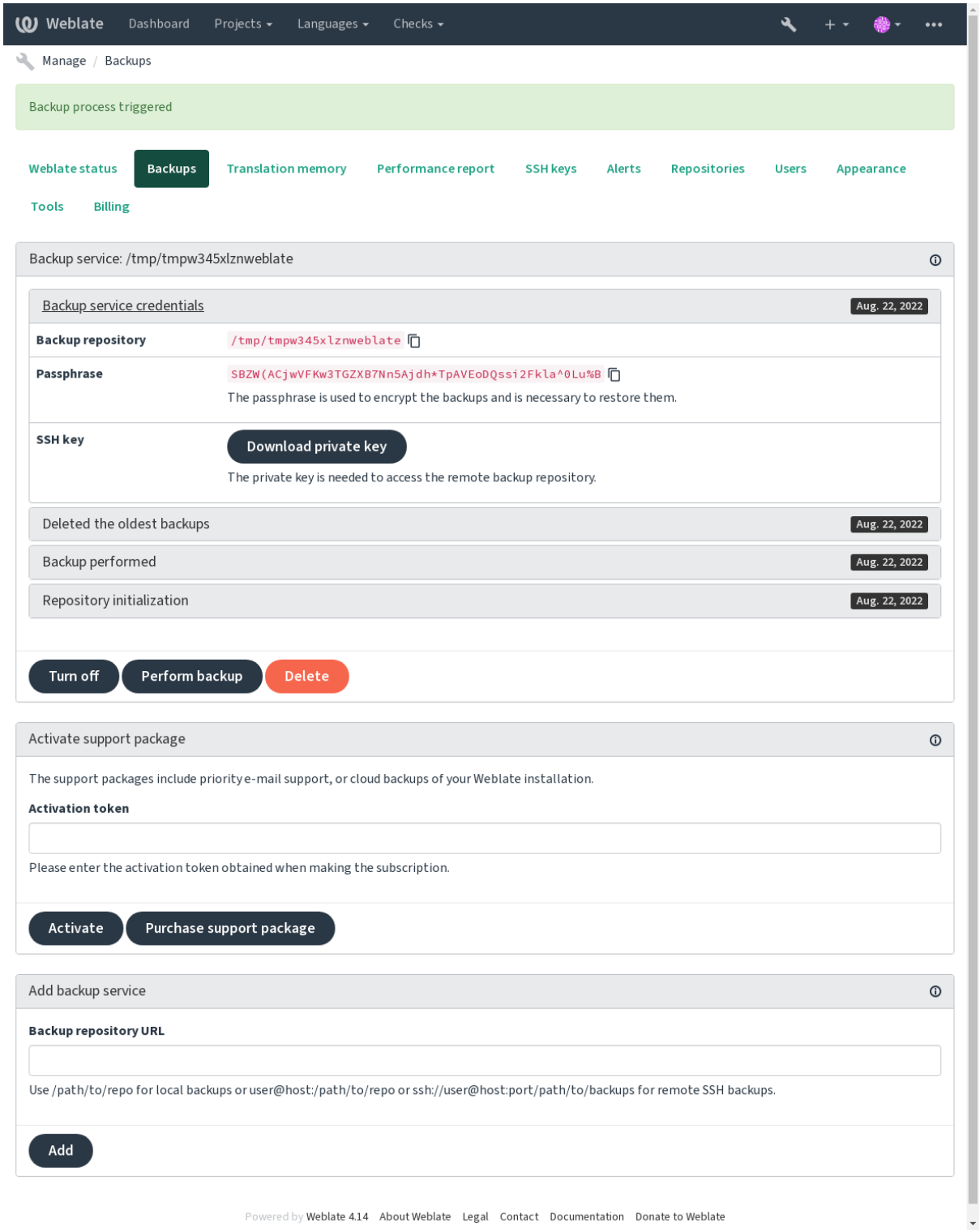
Novo na versão 3.9.

O Weblate tem suporte embutido para criação de backups de serviços a usar BorgBackup. Borg cria backups criptografados eficazes em termos de espaço que podem ser armazenados com segurança na nuvem. Os backups podem ser controlados na interface de gestão da guia *Backups*.

Alterado na versão 4.4.1: Os bancos de dados PostgreSQL e MySQL/MariaDB estão incluídos nos backups automatizados.

Os backups que usam o Borg são incrementais e o Weblate é configurado para manter os seguintes backups:

- Backups diários para 14 dias
- Backups semanais por 8 semanas atrás
- Backups mensais por 6 meses atrás


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/tmpw345xlznweblate
Backup service credentials
Aug. 22, 2022
Backup repository
/tmp/tmpw345xlznweblate
Passphrase
SBZW(ACjwVFKw3TGZXB7Nn5Ajdh*TpAVEoDQssi2Fkla^0Lu%B
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
Aug. 22, 2022
Backup performed
Aug. 22, 2022
Repository initialization
Aug. 22, 2022
Turn off
Perform backup
Delete
Activate support package
The support packages include priority e-mail support, or cloud backups of your Weblate installation.
Activation token
Please enter the activation token obtained when making the subscription.
Activate
Purchase support package
Add backup service
Backup repository URL
Use /path/to/repo for local backups or user@host:/path/to/repo or ssh://user@host:port/path/to/backups for remote SSH backups.
Add
Powered by Weblate 4.14
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Chave de criptografia do Borg

BorgBackup cria backups criptografados e não conseguiria restaurá-los sem a palavra-passe. A palavra-passe é gerada ao adicionar novo serviço de backup e deve copiá-lo e mantê-lo num lugar seguro.

Se estiver a usar *Armazenamento de backup provisionado do Weblate*, faça backup da sua chave SSH privada também, pois ela é usada para acessar os seus backups.

Veja também:

borg init

Personalizando o backup

- O backup do banco de dados pode ser configurado via `DATABASE_BACKUP`.
- A criação do backup pode ser personalizada a usar `BORG_EXTRA_ARGS`.

2.4.3 Armazenamento de backup provisionado do Weblate

A forma mais fácil de fazer backup da sua instância do Weblate é comprar o serviço de backup em weblate.org. É assim que o faz funcionar:

1. Compre o *Serviço de backup* em https://weblate.org/support/#backup.
2. Insira a chave obtida na interface de gestão, veja *Integrando o apoio*.
3. Weblate se conecta ao serviço de nuvem e obtém informações de acesso para os backups.
4. Ative a nova configuração de backup a partir da guia *Backups*.
5. Faça backup das suas credenciais do Borg para conseguir restaurar os backups, veja *Chave de criptografia do Borg*.

Dica: O passo manual para ativar tudo está lá para sua segurança. Sem o seu consentimento, nenhum dado é enviado ao repositório de backup obtido através do processo de registo.

2.4.4 Usar armazenamento de backup personalizado

Também pode usar o seu próprio armazenamento para backups. SSH pode ser usado para armazenar cópias de segurança no destino remoto, o servidor de destino precisa do BorgBackup instalado.

Veja também:

General na documentação do Borg

Sistema de ficheiros local

Recomenda-se especificar o caminho absoluto para o backup local, por exemplo */caminho/para/backup*. O diretório deve poder ser escrito pelo utilizador a executar o Weblate (veja *Permissões do sistema de ficheiros*). Se ele não existir, o Weblate tenta criá-lo, mas precisa das permissões apropriadas para fazê-lo.

Dica: Ao executar o Weblate no Docker, certifique-se de que o local de backup seja exposto como um volume do contentor Weblate. Caso contrário, os backups serão descartados pelo Docker na reinicialização do seu contentor.

Uma opção é pôr backups num volume existente. Por exemplo, `/app/data/borgbackup`. Este é um volume existente no contentor.

Também pode adicionar um novo contentor para os backups no ficheiro de composição do Docker, por exemplo, a usar `/borgbackup`:

```
services:
  weblate:
    volumes:
      - /home/weblate/data:/app/data
      - /home/weblate/borgbackup:/borgbackup
```

O diretório onde os backups serão armazenados para serem possuídos por UID 1000, caso o contrário, Weblate não será capaz de escrever os backups lá.

Backups remotos

Para criar os backups remotos, terá que instalar o BorgBackup em outro servidor que seja acessível para sua implantação de Weblate via SSH a usar a chave SSH do Weblate:

1. Prepare um servidor onde os seus backups serão armazenados.
2. Instale o servidor SSH nele (receberá-o por padrão com a maioria das distribuições Linux).
3. Instale o BorgBackup nesse servidor; a maioria das distribuições Linux tem pacotes disponíveis (veja Installation).
4. Escolha um utilizador existente ou crie um novo que será usado para backup.
5. Adicione a chave SSH do Weblate ao utilizador para que o Weblate possa usar o SSH para o servidor sem uma palavra-passe (veja *Chave SSH do Weblate*).
6. Configure o local de backup no Weblate como `utilizador@host:/caminho/para/backups` ou `utilizador@host/caminho/para/backups`.

Dica: *Armazenamento de backup provisionado do Weblate* fornece backups remotos automatizados sem qualquer esforço.

Veja também:

Chave SSH do Weblate, General

2.4.5 Restaurar do BorgBackup

1. Restaurar o acesso ao repositório de backup e preparar a sua palavra-passe de backup.
2. Liste todos os backups no servidor a usar `borg list REPOSITÓRIO`.
3. Restaure o backup desejado para o diretório atual a usar `borg extract REPOSITÓRIO::PACOTE`.
4. Restaure o banco de dados do despejo de SQL posto no diretório `backup` no diretório de dados do Weblate (veja :ref:"backup-dumps").
5. Copie a configuração do Weblate (`backups/settings.py`, veja *Dados despejados para backups*) até o local correto, veja *Ajustar a configuração*.

   Ao usar o contentor Docker, o ficheiro de configurações já está incluído no contentor e deve restaurar as variáveis de ambiente originais. O ficheiro `environment.yml` pode ajudá-lo com isso (veja *Dados despejados para backups*).
6. Copie todo o diretório de dados restaurados para o local configurado por `DATA_DIR`.

   Ao usar contentores do Docker, coloque os dados num volume de dados, veja *Volumes de contentor Docker*.

   Por favor, assegurar que os ficheiros têm o proprietário e as permissões corretas, veja :ref: *file-permissions*.

A sessão dos Borg pode parecer com isso:

```
$ borg list /tmp/xxx
Enter passphrase for key /tmp/xxx:
2019-09-26T14:56:08                  Thu, 2019-09-26 14:56:08␣
↪[de0e0f13643635d5090e9896bdaceb92a023050749ad3f3350e788f1a65576a5]
$ borg extract /tmp/xxx::2019-09-26T14:56:08
Enter passphrase for key /tmp/xxx:
```

Veja também:

borg list, borg extract

2.4.6 Backup manual

Dependendo do que deseja gravar, faça backup do tipo de dados que o Weblate armazena em cada lugar.

Dica: Se estiver a fazer os backups manualmente, pode silenciar os avisos do Weblate sobre a falta de backups a adicionar `weblate.I028` para `SILENCED_SYSTEM_CHECKS` em `settings.py` ou *`WEBLATE_SILENCED_SYSTEM_CHECKS`* para o Docker.

```
SILENCED_SYSTEM_CHECKS.append("weblate.I028")
```

Banco de dados

O local de armazenamento real depende da configuração do seu banco de dados.

Dica: O banco de dados é o armazenamento mais importante. Configure backups regulares do seu banco de dados. Sem o banco de dados, todas as traduções são perdidas.

Backup nativo do banco de dados

A abordagem recomendada é gravar um despejo do banco de dados a usar ferramentas nativas, tais como **`pg_dump`** ou **`msqldump`**. Esta abordagem normalmente tem um desempenho melhor do que o backup do Django e restaura tabelas completas com todos os seus dados.

Pode restaurar esse backup numa versão mais nova do Weblate, ele executará todas as migrações necessárias ao executar em `migrate`. Consulte *Atualizando o Weblate* sobre informações mais detalhadas sobre como a atualizar entre as versões.

Backup do banco de dados do Django

Alternativamente, pode fazer backup do seu banco de dados a utilizar o comando `dumpdata` do Django. Dessa forma o backup é agnóstico de banco de dados e pode ser usado caso queira alterar o backend do banco de dados.

Antes de restaurar o banco de dados, precisa estar a usar exatamente a mesma versão do Weblate na qual o backup foi feito. Isso é necessário, pois a estrutura do banco de dados muda entre as versões e acabaria a corromper os dados de alguma forma. Depois de instalar a mesma versão, execute todas as migrações do banco de dados a usar `migrate`.

Depois disso, algumas entradas já serão criadas no banco de dados e as terá no backup do banco de dados também. A abordagem recomendada é apagar essas entradas manualmente a usar o shell de gestão (veja *Invocando comandos de gestão*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Ficheiros

Se tiver espaço de backup suficiente, basta fazer backup de todo o *DATA_DIR*. Esta é uma aposta segura, mesmo que inclua alguns ficheiro que não quer. As seções a seguir descrevem o que deve fazer backup e o que pode pular em detalhes.

Dados despejados para backups

Alterado na versão 4.7: O despejo do ambiente foi adicionado como `environment.yml` para ajudar na restauração nos ambientes Docker.

Armazenados em *DATA_DIR* `/backups`.

O Weblate despeja vários dados aqui e pode incluir esses ficheiros para backups mais completos. Os ficheiros são atualizados diariamente (requer um servidor de «beats» do Celery em execução, consulte *Tarefas de fundo a usar o Celery*). Atualmente, isto inclui:

- Configurações do Weblate como `settings.py` (existe também a versão expandida em `settings-expanded.py`).
- Backup de banco de dados PostgreSQL como `database.sql`.
- Despejo do ambiente como `environment.yml`.

Os backups do banco de dados são salvos como texto simples por padrão, mas eles também podem ser comprimidos ou totalmente ignorados a usar *DATABASE_BACKUP*.

To restore the database backup load it using database tools, for example:

```
psql --file=database.sql weblate
```

Repositórios de controle de versão

Armazenados em *DATA_DIR* `/vcs`.

Os repositórios de controle de versão contêm uma cópia dos seus repositórios upstream com alterações do Weblate. Se tiver o *Enviar ao submeter* ativado para todos os seus componentes de tradução, todas as alterações do Weblate são incluídas no upstream. Não há necessidade de fazer backup dos repositórios no lado do Weblate, pois eles podem ser clonados novamente a partir dos upstream locais sem perda de dados.

Chaves SSH e GPG

Armazenados em *DATA_DIR* `/ssh` e *DATA_DIR* `/home`.

Se está a usar chaves SSH ou GPG geradas pelo Weblate, deve fazer backup destes locais. Caso contrário, vai perder as chaves privadas e terá que gerar novamente as novas.

Ficheiros enviados pelo utilizador

Armazenados em `DATA_DIR`/media.

Deve fazer o backup de todos os ficheiros enviados pelo utilizador (por exemplo, *Contexto visual para cadeias*).

Tarefas do Celery

A fila de tarefas do Celery pode conter algumas informações, mas geralmente não é necessária para um backup. No máximo, perderá atualizações ainda não processadas para a memória de tradução. Recomenda-se realizar a atualização de texto completo ou repositório ao restaurar de qualquer maneira, de modo que não há problema em perdê-las.

Veja também:

Tarefas de fundo a usar o Celery

Linha de comando para backup manual

Usando uma tarefa de cron, pode configurar um comando do Bash para ser executado diariamente, por exemplo:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz␣
↪backups vcs ssh home media fonts secret
```

A cadeia entre as aspas após *XZ_OPT* permite que escolha as suas opções do xz, por exemplo, a quantidade de memória utilizada para compressão; veja https://linux.die.net/man/1/xz

Pode ajustar a lista de pastas e ficheiros às suas necessidades. Para evitar gravar a memória de tradução (na pasta backups), pode usar:

```
$ XZ_OPT="-9" tar -Jcf ~/backup/weblate-backup-$(date -u +%Y-%m-%d_%H%M%S).xz␣
↪backups/database.sql backups/settings.py vcs ssh home media fonts secret
```

2.4.7 Restaurar backup manual

1. Restaure todos os dados dos quais tenha feito backup.
2. Atualize todos repositórios a usar o `updategit`.

   ```
   weblate updategit --all
   ```

2.4.8 Mover uma instalação do Weblate

Realoque a instalação de um sistema diferente, a seguir as instruções de backup e restauração acima.

Veja também:

Atualização de Python 2 para Python 3, *Migrando de outros bancos de dados para o PostgreSQL*

2.5 Autenticação

2.5.1 Registo de utilizador

A configuração predefinida para Weblate é usar python-social-auth, um formulário no site para lidar com o registo de novos utilizadores. Depois de confirmar o seu e-mail, um novo utilizador pode contribuir ou autenticar a usar um dos serviços de terceiros.

Também pode desativar o registo de novos utilizadores configurando `REGISTRATION_OPEN`.

As tentativas de autenticação estão sujeitas a *Limitação de taxa*.

2.5.2 Backends de autenticação

A solução embutida do Django é utilizada para autenticação, incluindo várias opções sociais para fazê-lo. Utilizando-a, pode importar o banco de dados de utilizadores de outros projetos baseados no Django (veja *Migrando de Pootle*).

Django pode, adicionalmente, ser configurado para autenticar em outros meios também.

Veja também:

Configurações de autenticação descreve como configurar a autenticação na imagem oficial do Docker.

2.5.3 Autenticação social

Graças ao Welcome to Python Social Auth's documentation!, o Weblate tem suporte a autenticação utilizando muitos serviços de terceiros, tais como GitLab, Ubuntu, Fedora, etc.

Por favor, verifique a documentação deles por instruções de configuração genéricas em Django Framework.

Nota: Por predefinição, o Weblate conta com serviços de autenticação de terceiros para fornecer um endereço de e-mail validado. Se alguns dos serviços que deseja usar não suportarem isto, por favor aplique a validação de e-mail no lado Weblate configurando FORCE_EMAIL_VALIDATION para eles. Por exemplo:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Veja também:

Pipeline

Permitir backends individuais é bastante fácil, é apenas uma questão de adicionar uma entrada à configuração `AUTHENTICATION_BACKENDS` e possivelmente adicionar chaves necessárias para um determinado método de autenticação. Por favor, note que alguns backends não fornecem e-mails do utilizador por predefinição, tem que solicitá-lo explicitamente, caso contrário o Weblate não será capaz de corretamente dar mérito às contribuições que os utilizadores fazem.

Dica: A maioria dos backends de autenticação exige HTTPS. Assim que o HTTPS estiver ativado no seu servidor web, configure o Weblate para relatá-lo corretamente usando `ENABLE_HTTPS` ou, no contentor Docker, `WEBLATE_ENABLE_HTTPS`.

Veja também:

Backend de Python Social Auth

Autenticação por OpenID

Para serviços baseados em OpenID, geralmente é apenas uma questão de ativá-los. A secção a seguir permite a autenticação OpenID para OpenSUSE, Fedora e Ubuntu:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.suse.OpenSUSEOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Veja também:

OpenID

Autenticação por GitHub

Precisa registar uma aplicação de OAuth no GitHub e, em seguida, dizer ao Weblate todos os seus segredos:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

O GitHub deve ser configurado para ter URL de um retorno de chamada como `https://example.com/accounts/complete/github/`.

Existem backends de autenticação semelhantes para o GitHub para organizações e o GitHub para equipas. As configurações delas são denominadas `SOCIAL_AUTH_GITHUB_ORG_*` e `SOCIAL_AUTH_GITHUB_TEAM_*` e requerem configuração adicional do escopo – `SOCIAL_AUTH_GITHUB_ORG_NAME` ou `SOCIAL_AUTH_GITHUB_TEAM_ID`. Os URLs de retorno delas de chamada são `https://example.com/accounts/complete/github-org/` e `https://example.com/accounts/complete/github-teams/`.

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

GitHub

Autenticação por Bitbucket

Precisa registar uma aplicação no Bitbucket e dar todos os segredos dele ao Weblate:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.BitbucketOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = "Bitbucket Client ID"
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = "Bitbucket Client Secret"
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

Bitbucket

OAuth 2 do Google

Para usar o OAuth 2 do Google, precisa registar-se numa aplicação em <https://console.developers.google.com/> e ativar a API do Google+.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

Google

OAuth 2 do Facebook

Como de costume com os serviços OAuth 2, precisa registar a sua aplicação no Facebook. Uma vez feito, pode configurar o Weblate para usá-lo:

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

Facebook

OAuth 2 do GitLab

Para usar o OAuth 2 do GitLab, precisa registar uma aplicação em <https://gitlab.com/profile/applications>.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/gitlab/` e garantir que marque o escopo *read_user*.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]

# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

GitLab

Active Directory do Microsoft Azure

Weblate pode ser configurado para usar inquilinos comuns ou específicos para autenticação.

O URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/azuread-oauth2/` para autenticação comum e `https://SERVIDOR WEBLATE/accounts/complete/azuread-tenant-oauth2/` para autenticação específica do inquilino.

```
# Azure AD common

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread.AzureADOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET = ""
```

```
# Azure AD Tenant

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread_tenant.AzureADTenantOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

Microsoft Azure Active Directory

Slack

Para usar o OAuth 2 do Slack, precisa registar uma aplicação em <https://api.slack.com/apps>.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/slack/`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.slack.SlackOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SLACK_KEY = ""
SOCIAL_AUTH_SLACK_SECRET = ""
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de obter erros sobre incompatibilidade de URL, pode corrigir isso, consulte *Definir domínio correto do site*.

Veja também:

Slack

A substituir nomes e ícones de métodos de autenticação

Pode substituir o nome de exibição do método de autenticação e o ícone a usar configurações como `SOCIAL_AUTH_<NOME>_IMAGE` e `SOCIAL_AUTH_<NOME>_TITLE`. Por exemplo, substituir a nomenclatura para Auth0 ficaria assim:

```
SOCIAL_AUTH_AUTH0_IMAGE = "custom.svg"
SOCIAL_AUTH_AUTH0_TITLE = "Custom auth"
```

Desativar autenticação por palavra-passe

Autenticação por e-mail e palavra-passe pode ser desativada através da remoção de `social_core.backends.email.EmailAuth` de `AUTHENTICATION_BACKENDS`. Mantenha sempre `weblate.accounts.auth.WeblateUserBackend` lá, pois é necessário para a funcionalidade central do Weblate.

Disabling e-mail authentication will disable all e-mail related functionality – user invitation or password reset feature.

Dica: You can still use password authentication for the admin interface, for users you manually create there. Just navigate to `/admin/login/`.

Por exemplo, a autenticação a usar apenas o provedor Open ID do openSUSE pode ser alcançada a usar o seguinte:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Autenticação por palavra-passe

A predefinição `settings.py` vem com um razoável conjunto de `AUTH_PASSWORD_VALIDATORS`:

- As palavras-passe não podem ser muito similares com as suas outras informações pessoais.
- As palavras-passe devem conter no mínimo de 10 caracteres.
- As palavras-passe não podem ser palaras-passe comumente usadas.
- As palavras-passe não podem ser inteiramente numéricas.
- As palavras-passe não podem consistir num único caractere ou apenas espaço em branco.
- As palavras-passe não podem corresponder a uma palavra-passe que já usou no passado.

Pode personalizar esta configuração para corresponder à sua política de palavra-passe.

Além disso, também pode instalar o django-zxcvbn-password o que dá bastante estimativas realistas de complexidade da palavra-passe e permite rejeitar palavras-passe abaixo de um determinado limite.

2.5.5 Autenticação por SAML

Novo na versão 4.1.1.

Siga as instruções do Python Social Auth para configuração. Diferenças notáveis:

- Weblate tem suporte a único IDP que tem de ser chamado de `weblate` em `SOCIAL_AUTH_SAML_ENABLED_IDPS`.
- A URL de metadados XML de SAML é `/accounts/metadata/saml/`.
- As configurações a seguir são preenchidas automaticamente: `SOCIAL_AUTH_SAML_SP_ENTITY_ID`, `SOCIAL_AUTH_SAML_TECHNICAL_CONTACT`, `SOCIAL_AUTH_SAML_SUPPORT_CONTACT`

Exemplo de configuração:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SAML_SP_ENTITY_ID = f"https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbnl+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}
```

A configuração padrão extrai detalhes do utilizador dos seguintes atributos, configure o seu IDP para fornecê-los:

Atributo	Referência de URI SAML
Nome completo	`urn:oid:2.5.4.3`
Primeiro nome	`urn:oid:2.5.4.42`
Nome familiar	`urn:oid:2.5.4.4`
E-mail	`urn:oid:0.9.2342.19200300.100.1.3`
Nome de utilizador	`urn:oid:0.9.2342.19200300.100.1.1`

Dica: The example above and the Docker image define an IDP called `weblate`. You might need to configure this string as *Relay* in your IDP.

Veja também:

Configurando SAML no Docker, SAML

2.5.6 Autenticação por LDAP

A autenticação por LDAP pode ser melhor alcançada utilizando o pacote *django-auth-ldap*. Pode instalá-lo através dos meios habituais:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

Dica: Este pacote está incluído no contentor Docker, veja *Instalando a usar Docker*.

Nota: Há algumas incompatibilidades no módulo Python LDAP 3.1.0, o que o pode impedir de usar essa versão. Se obter o erro AttributeError: "module" object has no attribute "_trace_level", fayendo o downgrade para python-ldap 3.0.0 pode ajudar.

Uma vez que tenha o pacote instalado, pode ligá-lo à autenticação do Django:

```
# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"

# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'

# List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
    # Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
    # 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
    "email": "mail",
}

# Hide the registration form
REGISTRATION_OPEN = False
```

Nota: Deve remover `'social_core.backends.email.EmailAuth'` da configuração `AUTHENTICATION_BACKENDS`, caso contrário, os utilizadores poderão definir a palavra-passe deles no Weblate e autenticar a usar-a. Manter `'weblate.accounts.auth.WeblateUserBackend'` ainda é necessário para fazer permissões e facilitar utilizadores anónimos. Também permitirá que faça login a usar uma conta administrativa local, se a criou (por exemplo, a usar `createadmin`).

Usando palavra-passe associada

Se não puder usar a associação direta para autenticação, precisará usar a pesquisa e fornecer um utilizador para associar à pesquisa. Por exemplo:

```
import ldap
from django_auth_ldap.config import LDAPSearch

AUTH_LDAP_BIND_DN = ""
AUTH_LDAP_BIND_PASSWORD = ""
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "ou=users,dc=example,dc=com", ldap.SCOPE_SUBTREE, "(uid=%(user)s)"
)
```

Integração com o Active Directory

```
import ldap
from django_auth_ldap.config import LDAPSearch, NestedActiveDirectoryGroupType

AUTH_LDAP_BIND_DN = "CN=ldap,CN=Users,DC=example,DC=com"
AUTH_LDAP_BIND_PASSWORD = "password"

# User and group search objects and types
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "CN=Users,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s)"
)

# Make selected group a superuser in Weblate
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "OU=Groups,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True
```

Veja também:

Django Authentication Using LDAP, Authentication

2.5.7 Autenticação por CAS

A autenticação por CAS pode ser alcançada a usar um pacote como o *django-cas-ng*.

O primeiro passo é divulgar o campo de e-mail do utilizador via CAS. Isso tem que ser configurado no próprio servidor CAS e requer que utilize pelo menos CAS v2, já que o CAS v1 não tem suporte a atributos.

O segundo passo é atualizar a Weblate para utilizar o seu servidor CAS e os seus atributos.

Para instalar *django-cas-ng*:

```
pip install django-cas-ng
```

Uma vez que o pacote está instalado, pode conectá-lo ao sistema de autenticação do Django a modificar o ficheiro `settings.py`:

```
# Add CAS backed, keep the Django one if you want to be able to sign in
# even without LDAP for the admin account
AUTHENTICATION_BACKENDS = (
    "django_cas_ng.backends.CASBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# CAS server address
CAS_SERVER_URL = "https://cas.example.net/cas/"

# Add django_cas_ng somewhere in the list of INSTALLED_APPS
INSTALLED_APPS = (..., "django_cas_ng")
```

Finalmente, um sinal pode ser usado para mapear o campo de e-mail para o objeto do utilizador. Para que isso funcione, tem que importar o sinal do pacote *django-cas-ng* e conectar o seu código com este sinal. Fazer isto em configurações de ficheiro pode causar problemas, portanto, é sugerido pôr-lo:

- No método `django.apps.AppConfig.ready()` da configuração do seu app
- No ficheiro `urls.py` do projeto (quando não há modelos)

```
from django_cas_ng.signals import cas_user_authenticated
from django.dispatch import receiver


@receiver(cas_user_authenticated)
def update_user_email_address(sender, user=None, attributes=None, **kwargs):
    # If your CAS server does not always include the email attribute
    # you can wrap the next two lines of code in a try/catch block.
    user.email = attributes["email"]
    user.save()
```

Veja também:

Django CAS NG

2.5.8 Configurando autenticação por Django de terceiros

Geralmente, qualquer extensão de autenticação Django deve funcionar com Weblate. Basta seguir as instruções da extensão, lembrando de manter o backend do utilizador Weblate instalado.

Veja também:

Autenticação por LDAP, *Autenticação por CAS*

Normalmente, a instalação consiste em adicionar uma autenticação de backend a `django:AUTHENTICATION_BACKENDS`e a instalar uma app de autenticação (se houver) no :setting:`django:INSTALLED_APPS:

```
AUTHENTICATION_BACKENDS = (
    # Add authentication backend here
    "weblate.accounts.auth.WeblateUserBackend",
)

INSTALLED_APPS += (
    # Install authentication app here
)
```

2.6 Controlo de acesso

O Weblate vem com um sistema de privilégios fino para atribuir permissões ao utilizador para toda a instância ou num escopo limitado.

Alterado na versão 3.0: Antes do Weblate 3.0, o sistema de privilégios era baseado apenas no sistema de privilégios do Django, mas é agora especificamente construído para Weblate. Se for usado algo mais antigo, consulte a documentação para a versão específica que está a usar.

2.6.1 Controle de acesso personalizado

Se não está a administrar toda a instalação do Weblate e apenas tem acesso para gerir certos projetos (como no Hosted Weblate), as suas opções de gestão de controle de acesso são limitadas às seguintes configurações. Se não precisa de nenhuma configuração complexa, elas são suficientes para si.

Controlo de acesso ao projeto

Nota: Este recurso está indisponível para projetos que executam o plan Libre no Hosted Weblate.

Pode limitar o acesso do utilizador a projetos individuais a selecionar uma configuração diferente de *Controle de acesso*. As opções disponíveis são:

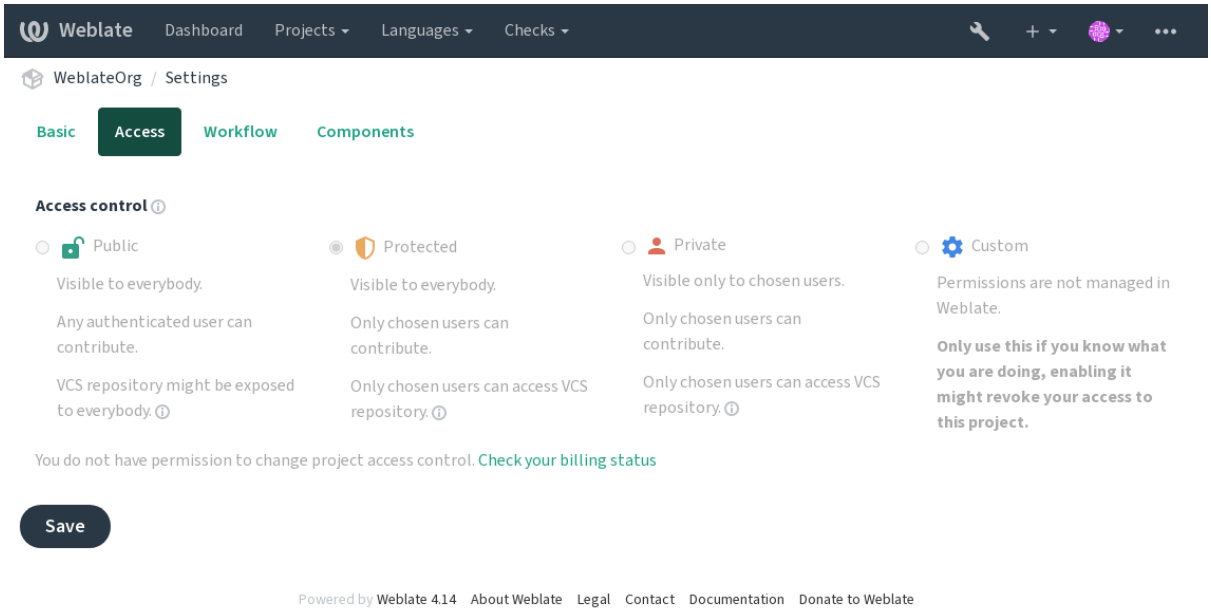
Pública
Visível publicamente, traduzível para todos os utilizadores autenticados.

Protegido
Publicamente visível, mas somente traduzível por utilizadores selecionados.

Privado
Visível e traduzível apenas por utilizadores selecionados.

Personalizado
Os recursos de *Gestão de utilizadores* serão desativados; por padrão, todos os utilizadores estão proibidos de realizar qualquer ação no projeto. Terá que configurar todas as permissões a usar *Controle de acesso personalizado*.

O *Controle de acesso* pode ser alterado na guia *Acesso* da configuração (*Gerir* ↓ *Configurações*) de cada projeto.



O valor padrão pode ser alterado por `DEFAULT_ACCESS_CONTROL`.

Nota: Mesmo para projetos *Privados*, algumas informações sobre o seu projeto serão expostas: estatísticas e resumo de linguagem para toda a instância incluirão contagens para todos os projetos, apesar da configuração de controle de acesso. O nome do seu projeto e outras informações não podem ser revelados por meio disso.

Nota: O conjunto real de permissões disponíveis para utilizadores por padrão em projetos *Públicos*, *Protegidos* e *Privados* pode ser redefinido pelo administrador da instância de Weblate a usar *configurações personalizadas*.

Aviso: Ao ativar o controle de acesso *Personalizado*, o Weblate removerá todos os *grupos especiais* que ele criou para um projeto selecionado. Se estiver a fazer isso sem permissão administrativa para toda a instância de Weblate, perderá instantaneamente o seu acesso para gerir o projeto.

Veja também:

Controlo de acesso

A gerir controle de acesso por projeto

Os utilizadores com o privilégio *gerir o acesso ao projecto* (ver *Lista de privilégios e funções internas*) podem gerir utilizadores em projetos através da adição deles às equipas. A coleção inicial de equipas é fornecida pela Weblate, mas equipas adicionais podem ser definidas, proporcionando um controlo de acesso mais fino. É possível limitar as equipas aos idiomas e atribuir-lhes funções de acesso designadas (ver *Lista de privilégios e funções internas*).

As equipas seguintes são criadas automaticamente para cada projeto:

Para projetos *Públicos*, *Protegidos* e *Privados*:

Administração
Inclui todas as permissões disponíveis para o projeto.

Revisar (somente se *fluxo de trabalho de revisão* estiver ativado)
Pode aprovar traduções durante a revisão.

Para projetos *Protegidos* e *Privados* apenas:

Traduzir
Pode traduzir o projeto e enviar traduções feitas offline.

Fontes
Pode editar cadeias fonte (se permitido nas *configurações do projeto*) e informações de cadeias fonte.

Idiomas
Pode gerir idiomas traduzidos (adicionar ou remover traduções).

Glossário
Pode gerir glossário (adicionar ou remover entradas e também enviar).

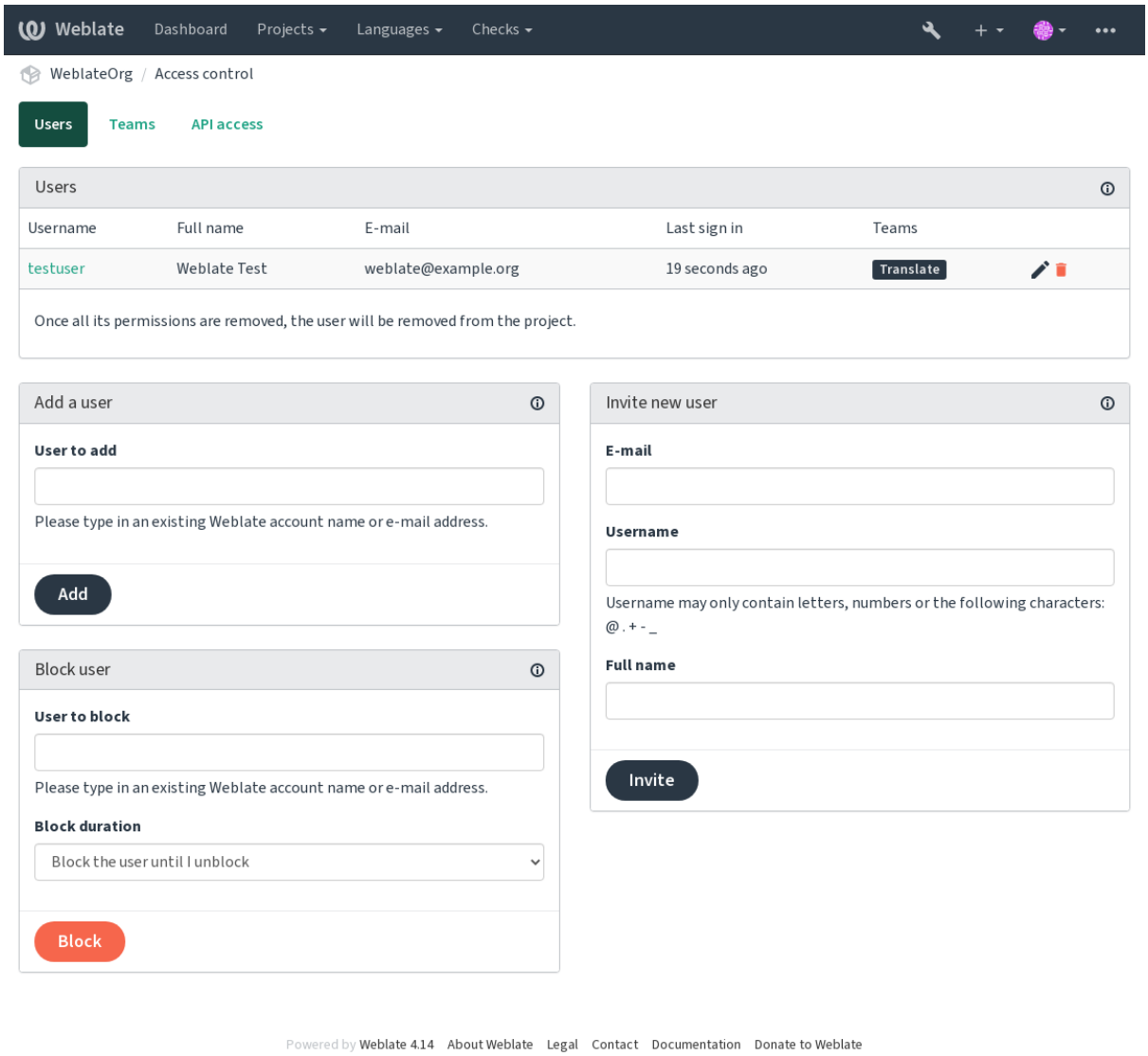
Memória
Pode gerir memória de tradução.

Capturas de ecrã
Pode gerir capturas de ecrã (adicioná-las ou removê-las e associá-las a cadeias fonte).

Tradução automática
Pode usar a tradução automática.

VCS
Pode gerir VCS e acessar o repositório exportado.

Faturação
Pode acessar informações de cobrança e configurações (consulte *Faturação*).



Esses recursos estão disponíveis na página *Controle de acesso*, que pode ser acessada no projeto a partir do menu *Gerir* ↓ *Utilizadores*.

Novo convite de utilizador

Além de adicionar um utilizador existente ao projeto, também é possível convidar novos. Qualquer novo utilizador será criado imediatamente, mas a conta permanecerá inativa até que entre com uma ligação no convite enviado por e-mail. Não é necessário ter nenhum privilégio de todo o site para fazer isso, a permissão de gestão de acesso no escopo do projeto (por exemplo, uma associação ao grupo *Administração*) seria suficiente.

Dica: Se o utilizador convidado perdeu a validade do convite, ele pode definir a sua palavra-passe a usar o endereço de e-mail convidado no formulário de redefinição de palavra-passe, pois a conta já foi criada.

Novo na versão 3.11: É possível reenviar o e-mail para convites de utilizador (a invalidar qualquer convite enviado anteriormente).

O mesmo tipo de convite está disponível em todo o site a partir da *interface de gestão* na guia *Utilizadores*.

A bloquear utilizadores

Novo na versão 4.7.

No caso de alguns utilizadores se comportarem mal no seu projeto, tem a opção de bloqueá-los de contribuir. O utilizador bloqueado ainda poderá ver o projeto se tiver permissão para isso, mas não poderá contribuir.

Gestão de permissões por projeto

Pode definir os seus projetos para *Protegido* ou *Privado* e *gerir utilizadores* por projeto na interface de utilizador do Weblate.

Por padrão, isso impede que o Weblate conceda acesso fornecido por *grupos padrão* *Utilizadores* e *Visualizadores* devido à própria configuração desses grupos. Isso não impede que conceda permissões a esses projetos em todo o site, a alterar grupos padrão, a criar um novo ou a criar configurações personalizadas adicionais para componentes individuais, conforme descrito em *Controle de acesso personalizado* abaixo.

Um dos principais benefícios de gerir permissões por meio da interface de utilizador do Weblate é que pode delegá-las a outros utilizadores sem lhes dar o privilégio de superutilizador. Para isso, adicione-os ao grupo *Administração* do projeto.

2.6.2 Controle de acesso personalizado

Nota: Este recurso está indisponível para projetos que executam o plan Libre no Hosted Weblate.

O sistema de permissão é baseado em grupos e funções, onde as funções de definir um conjunto de permissões e grupos vinculam-nas aos utilizadores e traduções. Veja *Utilizadores, papéis, grupos e permissões* para mais detalhes.

Os recursos mais poderosos do sistema de controle de acesso do Weblate por enquanto estão disponíveis apenas através da *interface administrativa do Django*. Pode usá-la para gerir as permissões de qualquer projeto. Não precisa necessariamente definir o *controle de acesso* para *Personalizado* para utilizá-lo. No entanto, deve ter privilégios de superutilizador para usá-lo.

Se não está interessado em detalhes de implementação e apenas deseja criar uma configuração simples o suficiente com base nos padrões, ou não tem um acesso de todo o site para toda a instalação do Weblate (como em Hosted Weblate), consulte a secção *Controle de acesso personalizado*.

Configurações comuns

Esta secção contém uma visão geral de algumas configurações comuns nas quais pode estar interessado.

Gestão de permissões para todo o site

Para gerir permissões para uma instância inteira de uma vez, adicione utilizadores aos *grupos padrão*:

- *Utilizadores* (isso é feito por padrão pela *associação de grupo automática*).
- *Revisores* (se estiver a usar *fluxo de trabalho de revisão* com revisores dedicados).
- *Gestores* (se deseja delegar a maioria das operações de gestão a outra pessoa).

Deve manter todos os projetos configurados como *Públicos* (veja *Controlo de acesso ao projeto*), caso contrário as permissões de todo o site fornecidas pela participação nos grupos *Utilizadores* e *Revisores* não terão nenhum efeito.

Também pode conceder algumas permissões adicionais de sua escolha aos grupos padrão. Por exemplo, pode dar permissão para gerir capturas de ecrã para todos os *Utilizadores*.

Também pode definir alguns novos grupos personalizados. Se deseja continuar a gerir as suas permissões em todo o site para esses grupos, escolha um valor apropriado para *Seleção de projeto* (por exemplo *Todos os projetos* ou *Todos os projetos públicos*).

Permissões personalizadas a idiomas, componentes ou projetos

Pode criar os seus próprios grupos dedicados para gerir permissões para objetos distintos, como linguagens, componentes e projetos. Embora esses grupos só possam conceder privilégios adicionais, não pode revogar qualquer permissão concedida por grupos de todo o site ou por projeto a adicionar outro grupo personalizado.

Exemplo:

Se quiser (por qualquer motivo) permitir a tradução para um idioma específico (digamos, *tcheco*) apenas para um conjunto fechado de tradutores confiáveis, a manter as traduções para outros idiomas públicos, terá que:

1. Remover a permissão para traduzir *Tcheco* de todos os utilizadores. Na configuração padrão, isso pode ser feito a alterar o *grupo padrão Utilizadores*.

Table 1: Grupo *Utilizadores*

Seleção de idioma	*Como definido*
Idiomas	Todos, exceto *Tcheco*

2. Adicionar um grupo dedicado para tradutores(as) de *Tcheco*.

Table 2: Grupo *Tradutores(as) de tcheco*

Funções	*Utilizadores(as) avançados(as)*
Seleção de projeto	*Todos os projetos públicos*
Seleção de idioma	*Como definido*
Idiomas	*Tcheco*

3. Adicionar utilizadores aos quais deseja conceder permissões a este grupo.

Como pode ver, a gestão de permissões dessa maneira é poderosa, mas pode ser um trabalho tedioso. Não pode delegá-lo a outro utilizador, a menos que conceda permissões de superutilizador.

Utilizadores, papéis, grupos e permissões

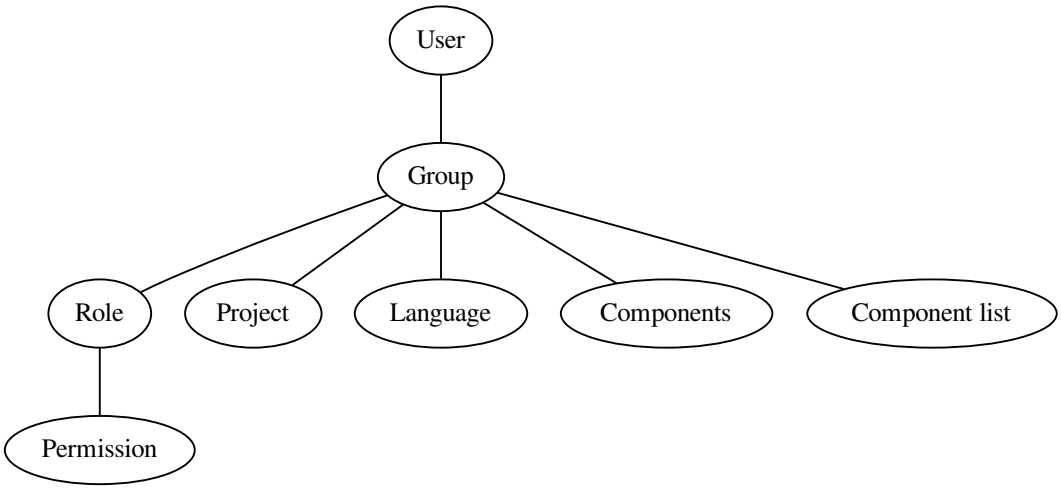
Os modelos de autenticação consistem em vários objetos:

Permissão
Permissão individual definida pelo Weblate. Permissões não podem ser atribuídas a utilizadores. Isso só pode ser feito através da atribuição de papéis.

Função
Um papel define um conjunto de permissões. Isso permite a reutilização desses conjuntos em vários lugares, a facilitar a administração.

Utilizador
Utilizador pode pertencer a vários grupos.

Grupo
Grupo conecta papéis, utilizadores e objetos de autenticação (projetos, idiomas e listas de componentes).



Nota: Um grupo não pode ter papéis lhe atribuídos; nesse caso, o acesso para navegar no projeto por qualquer pessoa é presumido (veja abaixo).

Acesso para navegar até um projeto

Um utilizador tem que ser um membro de um grupo vinculado ao projeto ou qualquer componente dentro daquele projeto. Ser membro é suficiente, não são necessárias permissões específicas para navegar o projeto (isso é usado no grupo padrão *Visualizadores*, consulte *Lista de grupos*).

Acesso para navegar até um componente

Um utilizador pode acessar componentes irrestritos assim que for possível acessar o projeto dos componentes (e terá todas as permissões que foram concedidas ao utilizador para o projeto). Com o *Acesso restrito* ativado, o acesso ao componente requer permissões explícitas para o componente (ou uma lista de componentes na qual o componente está contido).

Escopo de grupos

O escopo da permissão atribuída pelas funções nos grupos é aplicado pelas seguintes regras:

- Se o grupo especificar qualquer *Lista de componentes*, todas as permissões dadas aos membros desse grupo serão concedidas para todos os componentes nas listas de componentes anexadas ao grupo e um acesso sem permissões adicionais é concedido para todos os projetos estes componentes estão contidos. *Componentes* e *Projetos* são ignorados.
- Se o grupo especificar quaisquer *Componentes*, todas as permissões dadas aos membros desse grupo são concedidas para todos os componentes anexados ao grupo e um acesso sem permissões adicionais é concedido para todos os projetos nos quais que esses componentes estão. *Projetos* são ignorados.
- Caso contrário, se o grupo especificar *Projetos*, seja a listar-os diretamente ou a ter *Seleção de projetos* definida com um valor como *Todos os projetos públicos*, todas essas permissões são aplicadas a todos os projetos, que efetivamente concede as mesmas permissões para acessar todos os *componentes irrestritos* dos projetos.
- As restrições impostas pelos *Idiomas* de um grupo são aplicadas separadamente, quando é verificado se um utilizador tem acesso para realizar determinadas ações. Ou seja, é aplicado apenas a ações diretamente relacionadas ao próprio processo de tradução, como revisar, gravar traduções, adicionar sugestões, etc.

Dica: Use *Seleção de idioma* ou *Seleção de projeto* para automatizar a inclusão de todos os idiomas ou projetos.

Exemplo:

Digamos que haja um projeto `foo` com os componentes `foo/bar` e `foo/baz` e o seguinte grupo:

Table 3: Grupo *Revisores-admins de espanhol*

Funções	*Rever cadeias*, *Gerir repositório*
Componentes	foo/bar
Idiomas	*Espanhol*

Membros deste grupo terão as seguintes permissões (a presumir as configurações padrão de papeis):

- Acesso geral (navegação) para todo o projeto `foo` incluindo ambos componentes nele: `foo/bar` e `foo/baz`.
- Rever cadeias na tradução para espanhol de `foo/bar` (não em outro lugar).
- Gerir VCS para todo o repositório `foo/bar`, por exemplo, fazer commit de alterações pendentes feitas por tradutores de todos os idiomas.

Atribuições de grupo automáticas

Na parte inferior da página de edição *Grupo* na *interface administrativa do Django*, pode especificar *Associações automáticas de grupo*, que é uma lista de expressões regulares usadas para atribuir automaticamente utilizadores recém-criados a um grupo com base nos endereços de e-mail deles. Esta associação só acontece após a criação da conta.

O caso de uso mais comum para o recurso é atribuir todos os novos utilizadores a algum grupo padrão. Para fazer isso, provavelmente desejará manter o valor padrão (`^.*$`) No campo da expressão regular. Outro caso de uso para essa opção pode ser conceder alguns privilégios adicionais aos funcionários da sua empresa por padrão. Supondo que todos eles usem endereços de e-mail corporativos no seu domínio, isso pode ser feito com uma expressão como `^.*@minhaempresa.com`.

Nota: Associação automática de grupo para *Utilizadores* e *Visualizadores* sempre será recriada ao atualizar de uma versão anterior do Weblate para outra. Se deseja desativá-los, defina a expressão regular para `^$` (que nunca corresponderá).

Nota: Por enquanto, não há como adicionar em massa utilizadores já existentes a algum grupo por meio da interface do utilizador. Para isso, pode recorrer ao uso da *API REST*.

Grupos e funções predefinidos

Após a instalação, um conjunto padrão de grupos é criado (veja *Lista de grupos*).

Esses funções e grupos são criados após a instalação. As funções embutidas são sempre mantidas atualizadas pela migração do banco de dados ao atualizar. Não pode alterá-las. Defina uma nova função caso queira definir o seu próprio conjunto de permissões.

Lista de privilégios e funções internas

Âmbito	Permissão	Funções
Faturamento (consulte *Faturação*)	Ver informação de faturação	«Administração», «Faturação»
Alterações	Transferir alterações	*Administração*
Comentários	Publicar comentário	*Administração*, *Editar fonte*, *Utilizador avanç*
	Apagar comentário	*Administração*
	Resolver comentário	*Administração*, *Revisar cadeias*
Componente	Editar definições de componente	*Administração*
	Bloquear componente, impedindo traduções	*Administração*
Glossário	Adicionar entrada de glossário	*Administração*, *Gerir glossário*, *Utilizador ava*
	Editar entrada de glossário	*Administração*, *Gerir glossário*, *Utilizador ava*
	Apagar entrada de glossário	*Administração*, *Gerir glossário*, *Utilizador ava*
	Enviar entradas de glossário	*Administração*, *Gerir glossário*, *Utilizador ava*
Sugestões automáticas	Usar sugestões automáticas	*Administração*, *Editar fonte*, *Utilizador avanç*
Memória de tradução	Editar memória de tradução	*Administração*, *Gerir memória de tradução*
	Apagar memória de tradução	*Administração*, *Gerir memória de tradução*
Projetos	Editar definições do projeto	*Administração*
	Gerir acesso do projeto	*Administração*
Relatórios	Transferir relatórios	*Administração*
Capturas de ecrã	Adicionar captura de ecrã	*Administração*, *Gerir capturas de ecrã*
	Editar captura de ecrã	*Administração*, *Gerir capturas de ecrã*
	Apagar captura de ecrã	*Administração*, *Gerir capturas de ecrã*
Cadeias fonte	Editar informação adicional da cadeia	*Administração*, *Editar fonte*

Table 4 – continuação da página anterior

Âmbito	Permissão	Funções
Cadeias	Adicionar nova cadeia	*Administração*
	Remover uma cadeia	*Administração*
	Dispensar verificação com falha	*Administração*, *Editar fonte*, *Utilizador avanç*
	Editar cadeias	*Administração*, *Editar fonte*, *Utilizador avanç*
	Rever cadeias	*Administração*, *Revisar cadeias*
	Editar cadeia quando as sugestões são forçadas	*Administração*, *Revisar cadeias*
	Editar cadeias fonte	*Administração*, *Editar fonte*, *Utilizador avanç*
Sugestões	Aceitar sugestão	*Administração*, *Editar fonte*, *Utilizador avanç*
	Adicionar sugestão	*Administração*, *Editar fonte*, *Adicionar sugest*
	Apagar sugestão	*Administração*, *Utilizador avançado*
	Votar na sugestão	*Administração*, *Editar fonte*, *Utilizador avanç*
Traduções	Adicionar idioma para tradução	*Administração*, *Utilizador avançado*, *Gerir idi*
	Realizar tradução automática	*Administração*, *Tradução automática*
	Apagar tradução existente	*Administração*, *Gerir idiomas*
	Descarregar ficheiro de tradução	*Administração*, *Editar fonte*, *Acessar repositór*
	Adicionar vários idiomas para tradução	*Administração*, *Gerir idiomas*
Envios	Definir o autor da tradução enviada	*Administração*
	Substituir cadeias existentes com enviadas	*Administração*, *Editar fonte*, *Utilizador avanç*
	Enviar traduções	*Administração*, *Editar fonte*, *Utilizador avanç*
VCS	Aceder ao repositório interno	*Administração*, *Acessar repositório*, *Utilizador*
	Submeter alterações no repositório interno	*Administração*, *Gerir repositório*
	Enviar alteração do repositório interno	*Administração*, *Gerir repositório*
	Reiniciar alterações no repositório interno	*Administração*, *Gerir repositório*
	Ver localização do repositório principal	*Administração*, *Acessar repositório*, *Utilizador*
	Atualizar o repositório interno	*Administração*, *Gerir repositório*
Privilégios para todo o site	Utilizar a interface de gestão	
	Adicionar novos projetos	
	Adicionar definições de idioma	
	Gerir definições de idioma	
	Gerir grupos	
	Gerir utilizadores	
	Gerir funções	
	Gerir anúncios	
	Gerir a memória de tradução	
	Gerir tradução automática	
	Gerir as listas de componentes	

Nota: Privilégios para todo o site não são concedidos a nenhuma função padrão. Eles são poderosos e muito próximos do estado de superutilizador. A maioria deles afetam todos os projetos da sua instalação do Weblate.

Lista de grupos

Os seguintes grupos são criados após a instalação (ou após a execução de `setupgroups`) e está livre a modificá-los. A migração irá, no entanto, recriá-los se apagá-los ou renomeá-los.

Convidados

Define permissões para utilizadores não autenticados.

Este grupo contém apenas utilizadores anônimos (consulte `ANONYMOUS_USER_NAME`).

Pode remover funções deste grupo para limitar as permissões para utilizadores não autenticados.

Funções predefinidas: *Adicionar sugestão*, *Acessar repositório*

Visualizadores
Essa função garante a visibilidade de projetos públicos para todos os utilizadores. Por padrão, todos os utilizadores são membros deste grupo.

Por padrão, *associação automática de grupo* faz com que todos os novos membros de contas deste grupo quando eles entram.

Funções predefinidas: nenhuma

Utilizadores
Grupo predefinido para todos os utilizadores.

Por padrão, *associação automática de grupo* faz com que todos os novos membros de contas deste grupo quando eles entram.

Funções predefinidas: *Utilizador avançado*

Revisores
Grupo para revisores (consulte *Fluxos de trabalho de tradução*).

Funções predefinidas: *Revisar cadeias*

Gestores
Grupo pra administradores.

Funções predefinidas: *Administração*

Aviso: Nunca remova os grupos e utilizadores predefinidos do Weblate, pois isso pode levar a problemas inesperados! Se não tiver uso para eles, pode remover todos os privilégios deles.

2.6.3 Restrições adicionais de acesso

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 signing in to access all the site pages. This is basically the way to lock your Weblate installation.

Dica: You can use built-in *Novo convite de utilizador* to add new users.

2.7 Projetos de tradução

2.7.1 Organização de tradução

O Weblate organiza o conteúdo VCS traduzível do projeto/componentes numa estrutura semelhante a uma árvore.

- O objeto de nível inferior é *Configuração de projeto*, que deve conter todas as traduções pertencentes a um conjunto (por exemplo, tradução de uma aplicação em várias versões e/ou documentação que o acompanha).
- No nível acima, *Configuração de componente*, que é na verdade o componente a ser traduzido, define o repositório VCS a ser usado e a máscara dos ficheiros a serem traduzidos.
- Acima da *Configuração de componente* existem traduções individuais, manipuladas automaticamente pelo Weblate como ficheiros de tradução (que correspondem a *Máscara de ficheiros* definido em *Configuração de componente*) aparecem no repositório VCS.

O Weblate tem suporte a uma ampla gama de formatos de tradução (bilíngues e monolíngues) suportados pelo Translate Toolkit, veja *Formatos de ficheiros suportados*.

Nota: Pode compartilhar repositórios VCS clonados a usar *URLs internas do Weblate*. Usar este recurso é altamente recomendado quando tem muitos componentes a compartilhar o mesmo VCS. Ele melhora o desempenho e diminui o espaço em disco necessário.

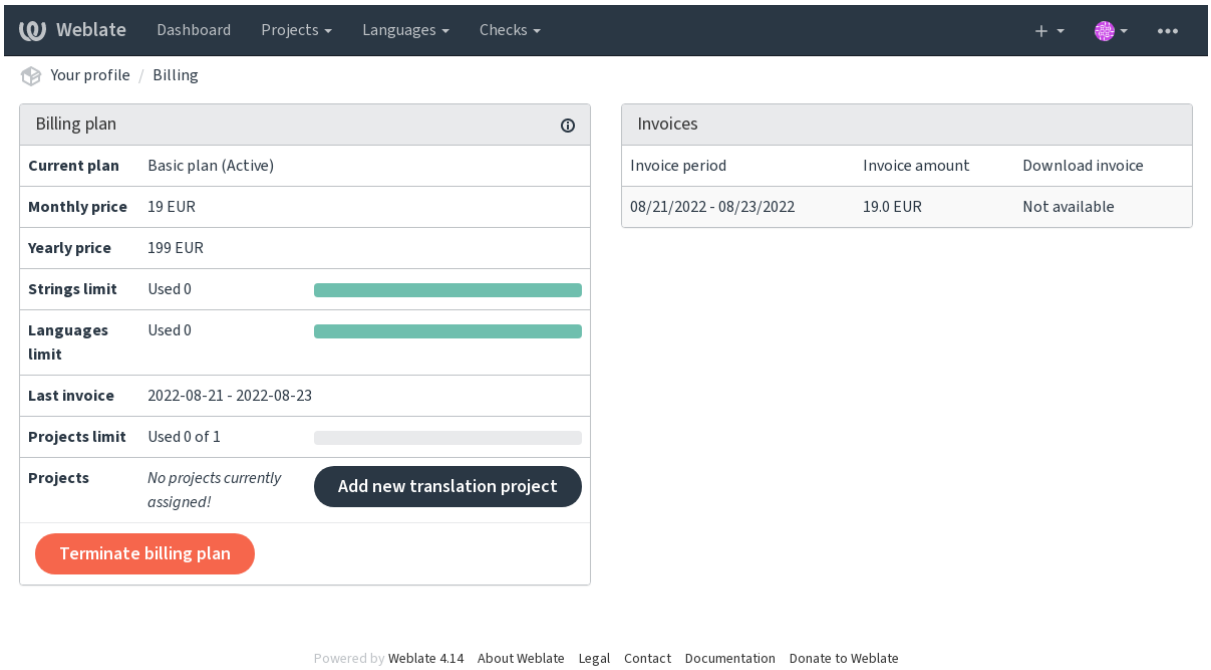
2.7.2 Adicionando projetos e componentes de tradução

Alterado na versão 3.2: Uma interface para adicionar projetos e componentes está incluída e não precisa mais usar *A interface administrativa do Django*.

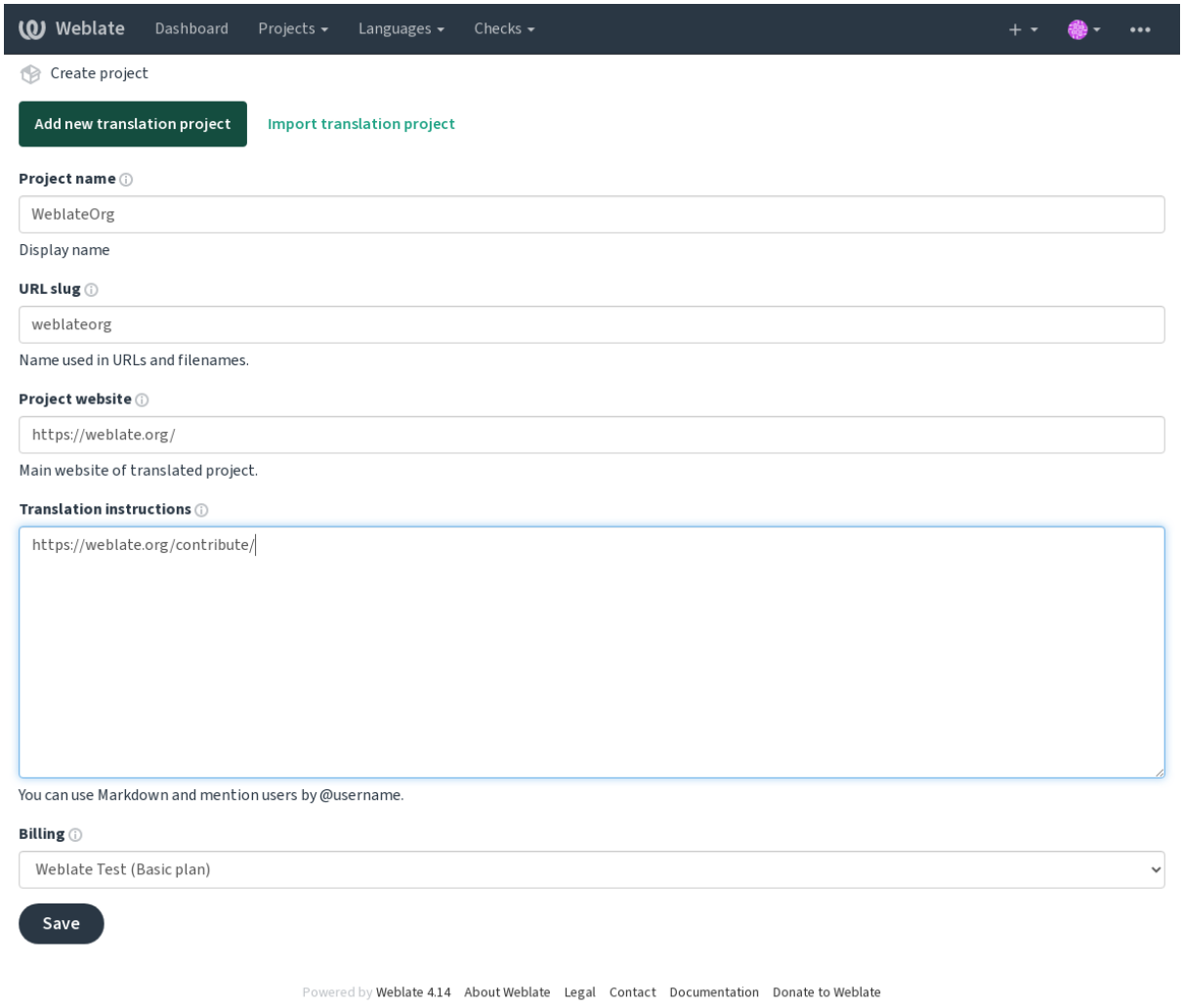
Alterado na versão 3.4: O processo de adição de componentes agora tem vários estágios, com descoberta automatizada da maioria dos parâmetros.

Com base nas suas permissões, novos projetos e componentes de tradução podem ser criados. É sempre permitido para utilizadores com a permissão *Adicionar novos projetos* e se a sua instância usa cobrança (por exemplo, como https://hosted.weblate.org/, veja *Faturação*), também pode criar aqueles com base na permissão dos seus planos da conta de utilizador que gere a cobrança.

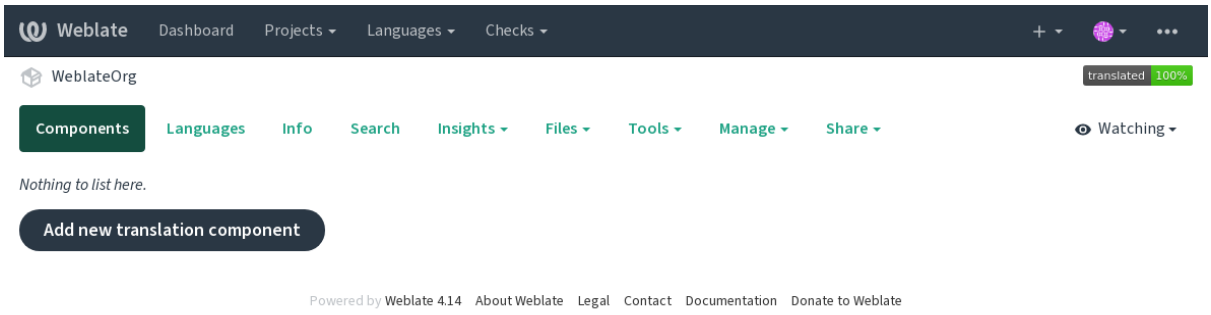
Pode visualizar o seu plano de cobrança atual numa página separada:



A criação do projeto pode ser iniciada a partir daí, ou a utilizar o menu da barra de navegação, a preencher informações básicas sobre o projeto de tradução para completar a adição dele:



Depois de criar o projeto, é levado diretamente à página do projeto:



A criação de um novo componente de tradução pode ser iniciada com um único clique lá. O processo de criação de um componente possui vários estágios e detecta automaticamente a maioria dos parâmetros de tradução. Existem várias abordagens para a criação de componentes:

Do controle de versão
Cria um componente do repositório de controle de versão remoto.

Do componente existente
Cria um componente adicional ao existente, a escolher ficheiros diferentes.

Ramo adicional
Cria um componente adicional ao existente, apenas para um ramo diferente.

Enviar ficheiros de tradução

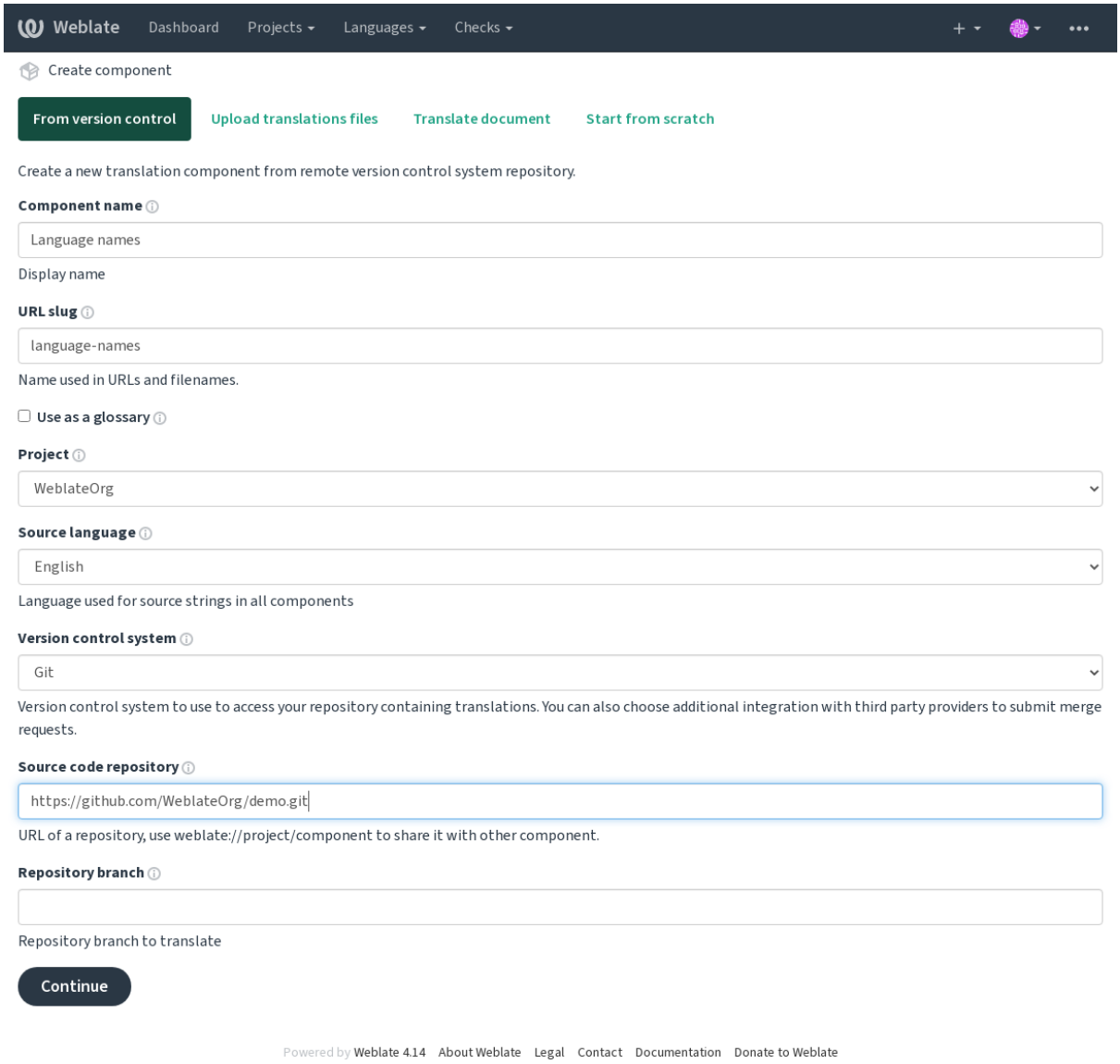
Envia os ficheiros de tradução para o Weblate caso não tenha controle de versão ou não queira integrá-lo ao Weblate. Pode atualizar o conteúdo posteriormente a usar a interface web ou *API REST do Weblate*.

Traduzir documento
Envia um único documento ou ficheiro de tradução e o traduz.

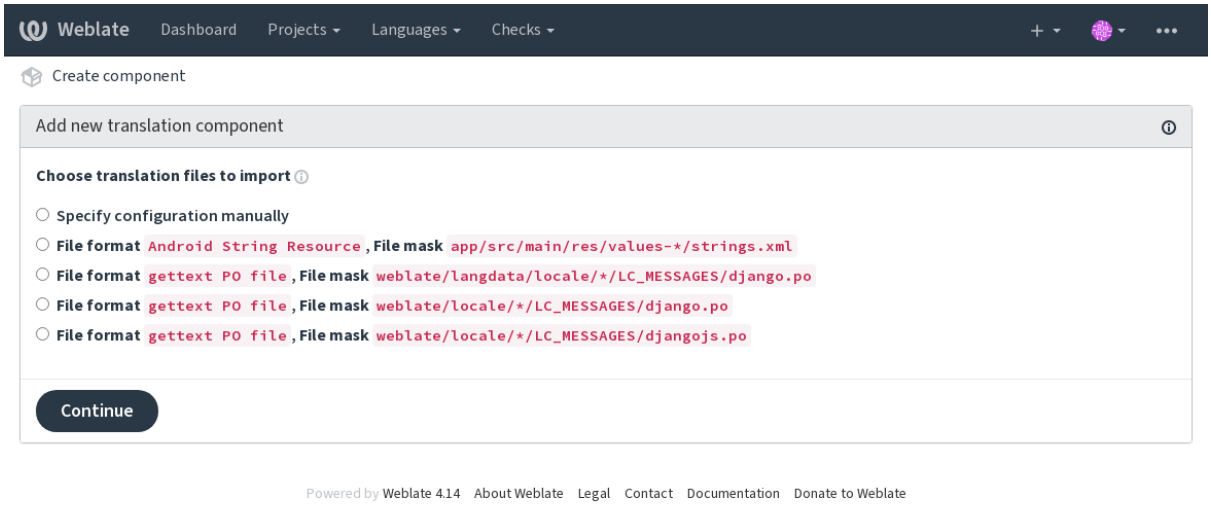
Começar do zero
Cria um projeto de tradução vazio e adiciona cadeias manualmente.

Depois de ter os componentes de tradução existentes, também pode adicionar facilmente novos para ficheiros adicionais ou ramos a usar o mesmo repositório.

Primeiro precisa preencher o nome e a localização do repositório:



Na próxima página, verá uma lista de recursos traduzíveis descobertos:



Como última etapa, analisa as informações do componente de tradução e preenche os detalhes opcionais:

Weblate Dashboard Projects Languages Checks

Create component

Add new translation component

Project

WeblateOrg

Component name

Language names

Display name

URL slug

language-names

Name used in URLs and filenames.

Version control system

Git

Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.

Source code repository

https://github.com/WeblateOrg/demo.git

URL of a repository, use weblate://project/component to share it with other component.

Repository branch

Repository branch to translate

Repository push URL

URL of a push repository, pushing is turned off if empty.

Push branch

Branch for pushing changes, leave empty to use repository branch

Repository browser

https://github.com/WeblateOrg/demo/blob/{{branch}}/{{filename}}#L{{line}}

Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. You might want to strip leading directory by using {{filename|parentdir}}.

File format

gettext PO file

File mask

app/src/main/res/values-*/strings.xmlweblate/langdata/locale/*/LC_MESSAGES/django.po

Path of files to translate relative to repository root, use * instead of language code, for example: po/*.po or locale/*/LC_MESSAGES/django.po.

Monolingual base language file

app/src/main/res/values/strings.xml

Filename of translation base file, containing all strings and their source; it is recommended for monolingual translation formats.

☑ **Edit base file**

Whether users will be able to edit the base file for monolingual translations.

Intermediate language file

Filename of intermediate translation file. In most cases this is a translation file provided by developers and is used when creating actual source strings.

Template for new translations

weblate/langdata/locale/django.pot

Filename of file used for creating new translations. For gettext choose .pot file.

Translation license

GNU General Public License v3.0 or later

Adding new translation

Create new language file

How to handle requests for creating new translations.

Language code style

Default based on the file format

Customize language code used to generate the filename for translations created by Weblate.

Language filter

^(cs|he|hu)$

Regular expression used to filter translation files when scanning for file mask.

Source language

English

Language used for source strings in all components

☐ **Use as a glossary**

You will be able to edit more options in the component settings after creating it.

Save

Powered by Weblate 4.14 About Weblate Legal Contact Documentation Donate to Weblate

Veja também:

A interface administrativa do Django, *Configuração de projeto*, *Configuração de componente*

2.7.3 Configuração de projeto

Cria um projeto de tradução e, em seguida, adiciona um novo componente para tradução nele. O projeto é como uma estante, na qual as traduções reais estão empilhadas. Todos os componentes no mesmo projeto compartilham sugestões e o dicionário deles; as traduções também são propagadas automaticamente por todos os componentes num único projeto (a menos que desligado na configuração do componente), veja *Memória de Tradução*.

Veja também:

/devel/integration

Esses atributos básicos configuram e informam os tradutores de um projeto:

Nome do projeto

Nome detalhado do projeto, usado para exibir o nome do projeto.

URL amigável

Nome do projeto adequado para URLs.

Site da Web do Projeto

URL onde os tradutores podem encontrar mais informações sobre o projeto.

Este é um parâmetro obrigatório, a menos que seja desligado por `WEBSITE_REQUIRED`.

Instruções para tradução

Texto que descreve o processo de localização no projeto e qualquer outra informação útil para tradutores. Markdown pode ser usado para a formatação de texto ou inserção de ligações.

Definir cabeçalho «Language-Team»

Se o Weblate deve gerir o cabeçalho `Language-Team` (este é um recurso apenas do *GNU gettext* no momento).

Utilizar memória de tradução partilhada

Para usar memória de tradução compartilhada, veja *Memória de tradução compartilhada* para mais detalhes.

The default value can be changed by `DEFAULT_SHARED_TM`.

Contribuir à memória de tradução compartilhada

Para contribuir com a memória de tradução compartilhada, veja *Memória de tradução compartilhada* para mais detalhes.

The default value can be changed by `DEFAULT_SHARED_TM`.

Controlo de acesso

Configura o controle de acesso por projeto, veja *Controlo de acesso ao projeto* para mais detalhes.

O valor padrão pode ser alterado por `DEFAULT_ACCESS_CONTROL`.

Activar revisões

Ativa o fluxo de trabalho de revisão para traduções, veja *Revisores dedicados*.

Ativar revisões de fontes

Ativa o fluxo de trabalho de revisão para cadeia fonte, veja *Revisões de cadeias fonte*.

Veja também:

report-source, *Comentários*

Ativar hooks

Se *Hooks de notificação* não autenticados devem ser usados para este repositório.

Veja também:

Ficheiro de idioma intermédio, *Portal de qualidade para cadeias fonte*, *Formatos bilíngues e monolíngues*, *Definições de idioma*

Aliases do idioma

Define o mapeamento dos códigos de idioma ao importar traduções para o Weblate. Use isso quando os códigos de idioma forem inconsistentes nos seus repositórios e quiser obter uma visualização consistente no Weblate ou no caso de desejar usar uma nomenclatura fora do padrão dos seus ficheiros de tradução.

O caso de uso típico pode ser mapear inglês americano para inglês: `en_US:en`

Vários mapeamentos devem ser separados por vírgula: `en_GB:en,en_US:en`

Usando código não padrão: `ia_FOO:ia`

Dica: Os códigos de idioma são mapeados ao combinar os ficheiros de tradução e as correspondências diferenciam maiúsculas de minúsculas, por isso certifique-se de usar os códigos de idioma fonte na mesma forma usada nos nomes dos ficheiros.

Veja também:

Analisando códigos de idioma

2.7.4 Configuração de componente

Um componente é um agrupamento de algo para tradução. Insere um local de repositório VCS e uma máscara de ficheiros para os ficheiros que deseja traduzir e o Weblate busca automaticamente neste VCS e encontra todos os ficheiros traduzíveis correspondentes.

Veja também:

/devel/integration

Pode encontrar alguns exemplos de configurações típicas nos *Formatos de ficheiros suportados*.

Nota: Recomenda-se manter os componentes de tradução num tamanho razoável – divida a tradução por qualquer coisa que faça sentido no seu caso (apps ou complementos individuais, capítulos de livros ou sites).

O Weblate lida facilmente com traduções com 10000s de cadeias, mas é mais difícil dividir o trabalho e coordenar entre os tradutores com componentes de tradução tão grandes.

Se a definição do idioma de uma tradução estiver a faltar, uma definição vazia é criada e nomeada como «cs_CZ (generated)». Deve ajustar a definição e relatar isso aos autores do Weblate, para que os idiomas em falta possam ser incluídos na próxima versão.

O componente contém todos os parâmetros importantes para trabalhar com o VCS e para obter traduções dele:

Nome do componente

Nome detalhado do componente, usado para exibir o nome do componente.

URL amigável do componente

Nome do componente adequado para URLs.

Projeto do componente

Configuração de projeto ao qual o componente pertence.

Sistema de controlo de versões

VCS para usar, veja *Integração de controlo de versões* para detalhes.

Veja também:

Fazendo push das alterações do Weblate

Repositório do código-fonte

Repositório VCS usado para fazer pull das alterações.

Veja também:

Veja *Acessando repositórios* para mais detalhes sobre especificar URLs.

Dica: Pode ser uma URL VCS real ou `weblate://projeto/componente` a indicar que o repositório deve ser compartilhado com outro componente. Veja *URLs internas do Weblate* para mais detalhes.

URL de submissão do repositório

URL do repositório usado para fazer push. Esta configuração é usada apenas para *Git* e *Mercurial* e o suporte a push é desativado para estes quando está vazio.

Para repositórios vinculados, isto não é usado e a definição do componente vinculado é aplicada.

Veja também:

Veja *Acessando repositórios* para mais detalhes sobre como especificar uma URL de repositório e *Fazendo push das alterações do Weblate* para mais detalhes sobre como fazer push de mudanças do Weblate.

Navegador do repositório

URL de navegação do repositório usada para exibir os ficheiros fonte (localização das mensagens usadas). Quando vazio, nenhuma dessas ligações será gerado. Pode usar *Marcação de modelo*.

Por exemplo, no GitHub, use algo como: `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename}}#L{{line}}`

No caso de os seus caminhos serem relativos a pastas diferentes (caminho contém `..`), pode remover o diretório principal pelo filtro `parentdir` (veja *Marcação de modelo*): `https://github.com/WeblateOrg/hello/blob/{{branch}}/{{filename|parentdir}}#L{{line}}`

URL do repositório exportado

URL para onde as alterações feitas pelo Weblate são exportadas. Isso é importante quando *Tradução contínua* não é usada, ou quando há a necessidade de mesclar manualmente as alterações. Pode usar *Exportador git* para automatizar isso para repositórios Git.

Ramo do repositório

De qual ramo fazer checkout do VCS e onde procurar traduções.

Para repositórios vinculados, isto não é usado e a definição do componente vinculado é aplicada.

Ramo do push

Ramo para fazer push de alterações, deixe vazio para usar *Ramo do repositório*.

Para repositórios vinculados, isto não é usado e a definição do componente vinculado é aplicada.

Nota: Atualmente, isso é compatível apenas com Git, GitLab e GitHub, mas é ignorado para outras integrações VCS.

Veja também:

Fazendo push das alterações do Weblate

Máscara de ficheiros

Máscara de ficheiros a traduzir, incluindo caminho. Deve incluir um «*» a substituir o código de idioma (veja *Definições de idioma* para informações sobre como isso é processado). Caso seu repositório contenha mais de um ficheiro de tradução (por exemplo, mais domínios gettext), precisa criar um componente para cada um deles.

Por exemplo, `po/*.po` ou `locale/*/LC_MESSAGES/django.po`.

No caso do seu nome de ficheiro conter caracteres especiais como `[`, `]`, eles precisam ser escapados como `[[]` ou `[]]`.

Veja também:

Formatos bilíngues e monolíngues, *O que significa «Existem mais ficheiros para o idioma único (en)»?*

Ficheiro de idioma base monolingue

Ficheiro base a conter definições de cadeias para *Componentes monolínguas*.

Veja também:

Formatos bilíngues e monolíngues, *O que significa «Existem mais ficheiros para o idioma único (en)»?*

Editar ficheiro base

Permite a edição do ficheiro base para *Componentes monolínguas*.

Ficheiro de idioma intermédio

Ficheiro de idioma intermediário para *Componentes monolínguas*. Na maioria dos casos, este é um ficheiro de tradução fornecido por programadores e é usado ao criar cadeias de origem reais.

When set, the source strings are based on this file, but all other languages are based on *Ficheiro de idioma base monolingue*. In case the string is not translated into the source language, translating to other languages is prohibited. This provides *Portal de qualidade para cadeias fonte*.

Veja também:

Portal de qualidade para cadeias fonte, *Formatos bilíngues e monolíngues*, *O que significa «Existem mais ficheiros para o idioma único (en)»?*

Modelo para novas traduções

Ficheiro base usado para gerar novas traduções, por exemplo, um ficheiro `.pot` com gettext.

Dica: In many monolingual formats Weblate starts with empty file by default. Use this in case you want to have all strings present with empty value when creating new translation.

Veja também:

adding-translation, *Adicionando novas traduções*, *Adicionar nova tradução*, *Formatos bilíngues e monolíngues*, *O que significa «Existem mais ficheiros para o idioma único (en)»?*

Formato de ficheiro

Formato do ficheiro de tradução, veja também *Formatos de ficheiros suportados*.

Endereço para reportar erros na cadeia fonte

Endereço de e-mail usado para relatar bugs upstream. Este endereço também receberá notificação sobre quaisquer comentários da cadeia fonte feitos no Weblate.

Permitir propagação da tradução

Pode desativar a propagação de traduções para este componente de outros componentes dentro do mesmo projeto. Isso realmente depende do que está a traduzir, às vezes é desejável ter feito uso de uma tradução mais de uma vez.

Normalmente, é uma boa ideia desligar isso para traduções monolíngues, a menos que esteja a usar os mesmos IDs em todo o projeto.

O valor padrão pode ser alterado por `DEFAULT_TRANSLATION_PROPAGATION`.

Veja também:

Manter traduções iguais entre componentes

Ativar sugestões

Se sugestões de tradução são aceitas para este componente.

Votação de sugestão

Ativa a emissão de votos para sugestões, veja *Votação de sugestão*.

Aceitar sugestões automaticamente

Aceita automaticamente sugestões votadas, veja *Votação de sugestão*.

Marcadores de tradução

Personalização de verificações de qualidade e outros comportamentos do Weblate, veja *Personalizando comportamento a usar marcadores*.

Verificações impostas

Lista de verificações que não podem ser ignoradas, veja *Forçar verificações*.

Nota: Forçar a verificação não a ativa automaticamente, ainda deve ativá-la *Personalizando comportamento a usar marcadores* em *Marcadores de tradução* ou *Informações adicionais sobre cadeias fonte*.

Licença da tradução

Licença da tradução (não precisa ser igual à licença do código-fonte).

Acordo de contribuidor

Acordo do utilizador que tem de ser aprovado antes do utilizador poder traduzir este componente.

Adicionar nova tradução

Como lidar com solicitações de criação de novos idiomas. Opções disponíveis:

Contactar gestores
O utilizador pode selecionar o idioma desejado e os mantenedores do projeto receberão uma notificação sobre isso. Cabe adiciona-los (ou não) o idioma ao repositório.

Apontar para URL com instruções de tradução
É apresentado ao utilizador uma ligação para a página que descreve o processo de iniciar novas traduções. Use-o caso deseje um processo mais formal (por exemplo, formar uma equipa antes de iniciar a tradução real).

Criar novo ficheiro de idioma
O utilizador pode selecionar o idioma e o Weblate cria automaticamente o ficheiro para ele e a tradução pode começar.

Desativar adição de novas traduções
Não haverá a opção para o utilizador iniciar uma nova tradução.

Dica: Os administradores do projeto podem adicionar novas traduções mesmo se estiver desativado aqui quando for possível (tanto *Modelo para novas traduções* ou o formato de ficheiro tem suporte a partir de um ficheiro vazio).

Veja também:

adding-translation, *Adicionando novas traduções*

Gerir cadeias

Novo na versão 4.5.

Configura se os utilizadores no Weblate terão permissão para adicionar novos textos e remover os existentes. Ajuste isso para corresponder ao seu fluxo de trabalho de localização – como os novos textos devem ser introduzidos.

Para formatos bilíngues, os textos são normalmente extraídos do código-fonte (por exemplo, usando **`xgettext`**) e adicionar novas strings no Weblate deve ser desativado (elas seriam descartadas na próxima vez que atualizasse os ficheiros de tradução). No Weblate, pode gerir textos para cada tradução e não exige que os textos em todas as traduções sejam consistentes.

Para formatos monolingues, as cadeias são geridas apenas no idioma fonte e são automaticamente adicionadas ou removidas nas traduções. As cadeias aparecem nos ficheiros de tradução assim que são traduzidas.

Veja também:

Formatos bilíngues e monolíngues, adding-new-strings, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Estilo de código de idioma

Personalizar o código de idioma usado para gerar o nome do ficheiro para traduções criadas por Weblate.

Veja também:

Adicionando novas traduções, *Código do idioma*, *Analisando códigos de idioma*

Estilo de união

Pode configurar como as atualizações a partir do repositório upstream são tratadas. A implementação real depende do VCS, veja :doc:`/vcs".

Re-basear
: Faz um rebase dos commits do Weblate no topo do repositório upstream ao atualizar. Isso fornece um histórico limpo sem commits de mesclagem adicionais.

  Rebasing pode causar problemas em caso de mesclagens complicadas, então considere cuidadosamente se quer ou não ativá-los.

  Pode precisar ativar o push forçado a escolher *Git com push forçado* como *Sistema de controlo de versões*, especialmente ao fazer push para um branch diferente.

Unir
: Upstream repository changes are merged into Weblate one. This setting utilizes fast-forward when possible. This is the safest way, but might produce a lot of merge commits.

Mesclar sem fast-forward
: Upstream repository changes are merged into Weblate one with doing a merge commit every time (even when fast-forward would be possible). Every Weblate change will appear as a merge commit in Weblate repository.

Valor predefindo pode ser alterado por `DEFAULT_MERGE_STYLE`.

Commit, add, delete, merge, add-on, and merge request messages

Mensagem usada ao fazer commit uma tradução, veja *Marcação de modelo*.

Default value can be changed by `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE`, `DEFAULT_MERGE_MESSAGE`, `DEFAULT_PULL_MESSAGE`.

Enviar ao submeter

Se as alterações confirmadas devem ser enviadas automaticamente para o repositório upstream. Quando ativado, o push é iniciado assim que o Weblate confirma as alterações no repositório subjacente dele (veja *Commits adiados*). Para ativar o push *URL de push do repositório* também deve ser configurado.

Idade das alterações a fazer commit

Define a idade (em horas) das alterações antes de ser feito commit pela tarefa em segundo plano ou pelo comando de gestão `commit_pending`. Todas as alterações num componente tem commit feito quando houver pelo menos uma alteração anterior a esse período.

O valor predefinido pode ser alterado por `COMMIT_PENDING_HOURS`.

Dica: Existem outras situações onde mudanças pendentes podem ter o commit feito, veja *Commits adiados*.

Bloquear com erro

Bloqueia o componente (e os componentes vinculados, consulte *URLs internas do Weblate*) no primeiro push ou mesclagem com falha no repositório upstream dele, ou extraia dele. Isso evita adicionar outros conflitos, que teriam que ser resolvidos manualmente.

O componente será desbloqueado automaticamente quando não houver mais erros de repositório.

Idioma fonte

Linguagem usada para textos fonte. Altere isso se estiver a traduzir de algo diferente do inglês.

Dica: Caso esteja a traduzir ficheiros bilíngues do inglês, mas também queira fazer correções na tradução para o inglês, escolha *Inglês (Programador)* como idioma fonte para evitar conflito entre o nome do idioma fonte e a tradução existente.

Para traduções monolíngues, pode usar tradução intermediária neste caso, veja *Ficheiro de idioma intermédio*.

Filtro de idioma

Expressão regular usada para filtrar a tradução ao examinar por uma máscara de ficheiros. Pode ser usada para limitar a lista de idiomas gerenciados pelo Weblate.

Nota: Precisa listar os códigos de idioma conforme aparecem no nome do ficheiro.

Alguns exemplos de filtragem:

Descrição do filtro	Expressão regular
Apenas idiomas selecionados	`^(cs|de|es)$`
Exclui idiomas	`^(?!(it|fr)$).+$`
Filtra apenas códigos de duas letras	`^..$`
Exclui ficheiros que não são idiomas	`^(?!(blank)$).+$`
Inclui todos os ficheiros (predfinição)	`^[^.]+$`

Expressão regular das variantes

Expressão regular usada para determinar as variantes de um texto, veja variants.

Nota: Most of the fields can be edited by project owners or administrators, in the Weblate interface.

Veja também:

O Weblate tem suporte a outros VCSs que não o Git e o Mercurial?, alerts

Prioridade

Componentes de prioridade mais elevada são oferecidos primeiro aos tradutores.

Acesso restrito

Por predefinição, o componente é visível para todos que tenham acesso ao projeto, mesmo se a pessoa não puder realizar nenhuma alteração no componente. Isso torna mais fácil manter a consistência da tradução dentro do projeto.

Restringir o acesso num componente ou nível de lista de componentes assume a permissão de acesso a um componente, independentemente das permissões de nível de projeto. Terá que conceder-lhe acesso explicitamente. Isso pode ser feito concedendo acesso a um novo grupo de utilizadores e colocando utilizadores nele, ou usando os grupos de controle de acesso predefinido *personalizado* ou *privado*.

O valor predefinido pode ser alterado em `DEFAULT_RESTRICTED_COMPONENT`.

Dica: Isso se aplica aos administradores do projeto também – certifique-se de não perder o acesso ao componente após alternar o status.

Compartilhar em projetos

Pode escolher projetos adicionais onde o componente ficará visível. Útil para bibliotecas compartilhadas que usa em vários projetos.

Nota: Compartilhar um componente não altera o controle de acesso dele. Só o torna visível ao navegar em outros projetos. Os utilizadores ainda precisam acessar o componente real para navegar ou traduzi-lo.

Usar como um glossário

Novo na versão 4.5.

Permite usar este componente como glossário. Pode configurar como ele será listado a usar a *Cor do glossário*.

O glossário estará acessível em todos os projetos definidos por *Compartilhar em projetos*.

É recomendado ativar *Gerir cadeias* em glossários para permitir adicionar-lhes novas palavras.

Veja também:

Glossário

Cor do glossário

Cor de exibição para um glossário usado ao mostrar correspondências de palavras.

2.7.5 Marcação de modelo

O Weblate usa uma linguagem de marcação simples em vários lugares onde a renderização de texto é necessária. É baseado em The Django template language, então pode ser bastante poderoso.

Atualmente isso é usado em:

- Formatação de mensagens de commit, veja *Configuração de componente*
- **Várias extensões**
  - *Descoberta de componentes*
  - *Gerador de estatísticas*
  - *Escrevendo scripts para extensões*

As seguintes variáveis estão disponíveis nos modelos de componentes:

`{{ 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 }}`
URL de tradução

`{{ filename }}`
Nome do ficheiro de tradução

`{{ stats }}`
Estatísticas de tradução, isso tem outros atributos, exemplos abaixo.

`{{ stats.all }}`
Contagem total de textos

`{{ stats.fuzzy }}`
Contagem de textos que precisam de revisão

`{{ stats.fuzzy_percent }}`
Percentagem de textos que precisam de revisão

`{{ stats.translated }}`
Contagem de textos traduzidos

`{{ stats.translated_percent }}`
Percentagem de textos traduzidos

`{{ stats.allchecks }}`
Quantidade de textos com falhas nas verificações

`{{ stats.allchecks_percent }}`
Percentagem de textos com falhas nas verificações

`{{ author }}`
Autor do commit atual, disponível apenas no escopo do commit.

{{ addon_name }}
Nome da extensão atualmente em execução, disponível apenas na mensagem de commit da extensão.

As seguintes variáveis estão disponíveis nos modelos de navegador ou editor de repositório:

{{branch}}
ramo atual

{{line}}
linha no ficheiro

{{filename}}
nome do ficheiro, também pode remover as partes principais usando o filtro `parentdir`, por exemplo `{{filename|parentdir}}`

Pode combiná-los com filtros:

```
{{ component|title }}
```

Pode usar as condições:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

Há uma tag adicional disponível para substituir caracteres:

```
{% replace component "-" " " %}
```

Pode combiná-lo com filtros:

```
{% replace component|capfirst "-" " " %}
```

Existem também filtros adicionais para manipular com nomes de ficheiros:

```
Directory of a file: {{ filename|dirname }}
File without extension: {{ filename|stripext }}
File in parent dir: {{ filename|parentdir }}
It can be used multiple times:  {{ filename|parentdir|parentdir }}
```

…e outros recursos de modelos do Django.

2.7.6 Velocidade de importação

Obter o repositório VCS e importar traduções ao Weblate pode ser um processo demorado, dependendo do tamanho das suas traduções. Aqui estão algumas dicas:

Otimizar a configuração

A configuração predefinida é útil para testar e depurar o Weblate, enquanto para uma configuração de produção, deve fazer alguns ajustes. Muitas delas têm um grande impacto no desempenho. Por favor, confira *Configuração de produção* para mais detalhes, especialmente:

- Configure o Celery para executar tarefas em segundo plano (veja *Tarefas de fundo a usar o Celery*)
- *Ativar o cache*
- *Usar um poderoso mecanismo de banco de dados*
- *Desativar o modo de depuração*

Verificar limites de recursos

Se estiver a importar grandes traduções ou repositórios, pode ser atingido por limitações de recursos do seu servidor.

- Verifique a quantidade de memória livre, pois os ficheiros de tradução armazenados em cache pelo sistema operacional melhorará muito o desempenho.
- As operações de disco podem ser um gargalo se houver muitos textos a serem processadas – o disco é enviado pelo Weblate e pelo banco de dados.
- Núcleos de CPU adicionais podem ajudar a melhorar o desempenho de tarefas em segundo plano (veja *Tarefas de fundo a usar o Celery*).

Desativar verificações desnecessárias

Algumas verificações de qualidade podem ser bastante custosas e, se não forem necessárias, podem economizar algum tempo durante a importação, se omitidas. Veja `CHECK_LIST` para informações sobre configuração.

2.7.7 Criação automática de componentes

Caso seu projeto tenha dezenas de ficheiros de tradução (por exemplo, para diferentes domínios gettext ou partes de aplicações Android), pode importá-los automaticamente. Isso pode ser feito na linha de comando usando `import_project` ou `import_json`, ou instalando a extensão *Descoberta de componentes*.

Para usar a extensão, primeiro precisa criar um componente para um ficheiro de tradução (escolha aquele que tem menos probabilidade de ser renomeado ou removido no futuro) e instalar a extensão neste componente.

Para os comandos de gestão, precisa criar um projeto que conterá todos os componentes e então executar `import_project` ou `import_json`.

Veja também:

Comandos de gerência, *Descoberta de componentes*

2.8 Definições de idioma

Para apresentar diferentes traduções corretamente, são necessárias informações sobre nome do idioma, direção de texto, definições plurais e código de idioma.

2.8.1 Definições embutidas de idioma

Definições para cerca de 600 idiomas estão incluídas no Weblate e a lista é estendida a cada versão. Sempre que o Weblate é atualizado (mais especificamente sempre que **`weblate migrate`** é executado, veja *Instruções genéricas de atualização*) o banco de dados de idiomas é atualizado para incluir todas as definições de idioma enviadas no Weblate.

Este recurso pode ser desativado a usar `UPDATE_LANGUAGES`. Também pode forçar a atualização do banco de dados para corresponder aos dados embutidos do Weblate a usar `setuplang`.

Veja também:

Estendendo definições embutidas de idioma, Current language definitions

2.8.2 Analisando códigos de idioma

Ao analisar as traduções, o Weblate tenta mapear o código de idioma (geralmente o ISO 639-1) a partir da :ref: *component-filemask* para qualquer objeto de idioma existente.

Pode ajustar ainda mais esse mapeamento no nível do projeto por *Aliases do idioma*.

Se nenhuma correspondência exata for encontrada, uma tentativa será feita para melhor encaixá-la num idioma existente. São testadas etapas a seguir:

- Procuras sem diferenciar maiúsculo de minúsculo.
- Normalizar sublinhados e traços.
- Procurar aliases de idioma embutido.
- Procurar por nome de idioma.
- Ignorar o código de país padrão para o idioma dado – escolher `cs` em vez de `cs_CZ`.

Se isso também falhar, uma nova definição de idioma será criada a usar seu padrão (direção do texto da esquerda para a direita, um plural). A linguagem criada automaticamente com o código `xx_XX` será nomeada como *xx_XX (generated)*. Podia mudar isso na interface administrativa mais tarde, (veja *Alterando as definições de idioma*) e relatar ao rastreador de problemas (veja *Contribuir ao Weblate*), para que a definição apropriada possa ser adicionada ao próximo lançamento do Weblate.

Dica: No caso de ver algo indesejado como idioma, podia ajustar *Filtro de idioma* para ignorar tal ficheiro ao analisar traduções.

Veja também:

Código do idioma, *Adicionando novas traduções*

2.8.3 Alterando as definições de idioma

Pode alterar as definições de idioma na interface de idiomas (URL `/languages/`).

Durante a edição, certifique-se de que todos os campos estejam corretos (especialmente os plurais e a direção do texto), caso contrário, os tradutores não poderão editar adequadamente essas traduções.

2.8.4 Códigos de idioma ambíguos e macrolinguagens

In many cases it is not a good idea to use macrolanguage code for a translation. The typical problematic case might be Kurdish language, which might be written in Arabic or Latin script, depending on actual variant. To get correct behavior in Weblate, it is recommended to use individual language codes only and avoid macrolanguages.

Veja também:

Definição de macroidiomas, Lista de macroidiomas

2.8.5 Definições de idioma

Cada idioma consiste nos seguintes campos:

Código do idioma

Código identificando o idioma. Weblate prefere códigos de duas letras, conforme definido pela ISO 639-1, mas usa códigos de ISO 639-2 or ISO 639-3 para idiomas que não têm código de duas letras. Ele também pode ter suporte a códigos estendidos, conforme definido por BCP 47.

Veja também:

Analisando códigos de idioma, *Adicionando novas traduções*

Nome do idioma

Nome visível do idioma. Os nomes dos idiomas incluídos no Weblate também estão a ser localizados a depender do idioma da interface do utilizador.

Direção do texto

Determina se o idioma é escrito da direita para a esquerda ou da esquerda para a direita. Esta propriedade é autodetectada corretamente para a maioria dos idiomas.

Quantidade de plurais

Quantidade de plurais usados no idioma.

Fórmula de plural

Fórmula de plural compatível com Gettext usada para determinar qual forma de plural é usada para determinada contagem.

Veja também:

Plurais, GNU gettext utilities: Plural forms, Language Plural Rules by the Unicode Consortium

2.8.6 Adicionando novas traduções

Alterado na versão 2.18: Nas versões anteriores a 2.18, o comportamento de adicionar novas traduções era específico do formato de ficheiro.

Weblate pode iniciar automaticamente uma nova tradução para todos os formatos de ficheiro.

Alguns formatos esperam começar com um ficheiro vazio e apenas textos traduzidos para serem incluídas (por exemplo, *Recurso de cadeias de Android*), enquanto outros esperam ter todas as chaves presentes (por exemplo, *GNU gettext*). Os formatos baseado em documentos (por exemplo, *Formato OpenDocument*) começam com um cópia vazia do documento fonte e todos textos marcados como a necessitar edição. Em algumas situações, isso realmente não depende do formato, mas sim do framework que usa para lidar com a tradução (por exemplo, com *Ficheiro JSON*).

Quando especifica *Modelo para novas traduções* em *Configuração de componente*, Weblate usará este ficheiro para iniciar novas traduções. Todas as traduções existentes serão removidas do ficheiro ao fazer isso.

Quando *Modelo para novas traduções* está vazio e o formato de ficheiro suporta, um ficheiro vazio é criado onde novas cadeias serão adicionadas assim que forem traduzidos.

O *Estilo de código de idioma* permite que personalize o código de idioma usado nos nomes de ficheiros gerados:

Predefinição baseada no formato do ficheiro
Dependendo do formato do ficheiro, para a maioria deles utiliza-se POSIX.

Estilo POSIX utilizando o sublinhado como um separador
Normalmente usado por gettext e ferramentas relacionadas, produz códigos de linguagem como `pt_BR`.

Estilo de POSIX utilizando o sublinhado como um separador, incluindo o código do país
Código de idioma do estilo POSIX incluindo o código do país mesmo quando não necessário (por exemplo, `cs_CZ`).

Estilo BCP utilizando o hífen como um separador
Normalmente usado em plataformas web, produz códigos de idioma como `pt-BR`.

Estilo de BCP utilizando o hífen como um separador, incluindo o código do país
Código de idioma do estilo BCP, incluindo o código do país, mesmo quando não necessário (por exemplo, `cs-CZ`).

BCP style using hyphen as a separator, legacy language codes
Uses legacy codes for Chinese and BCP style notation.

BCP style using hyphen as a separator, lower cased
BCP style notation, all in lower case (for example `cs-cz`).

App store metadata style
Style suitable for uploading metadata to appstores. This should be suitable for both Apple App Store and Google Play Store.

Estilo Android
Usado apenas em apps Android, produz códigos de idioma como `pt-rBR`.

Estilo Linux
Localidades como usado pelo Linux, usa códigos legados para notação de estilo chinês e POSIX.

Além disso, quaisquer mapeamentos definidos em *Aliases do idioma* são aplicados ao contrário.

Nota: O Weblate reconhece qualquer um desses ao analisar ficheiros de tradução; as configurações acima apenas influenciam como os novos ficheiros são criados.

Veja também:

Código do idioma, *Analisando códigos de idioma*

2.9 Tradução contínua

Há infraestrutura em vigor para que a sua tradução acompanhe o desenvolvimento de perto . Desta forma, os tradutores podem trabalhar em traduções o tempo todo, em vez de trabalhar com uma enorme quantidade de texto novo pouco antes do lançamento.

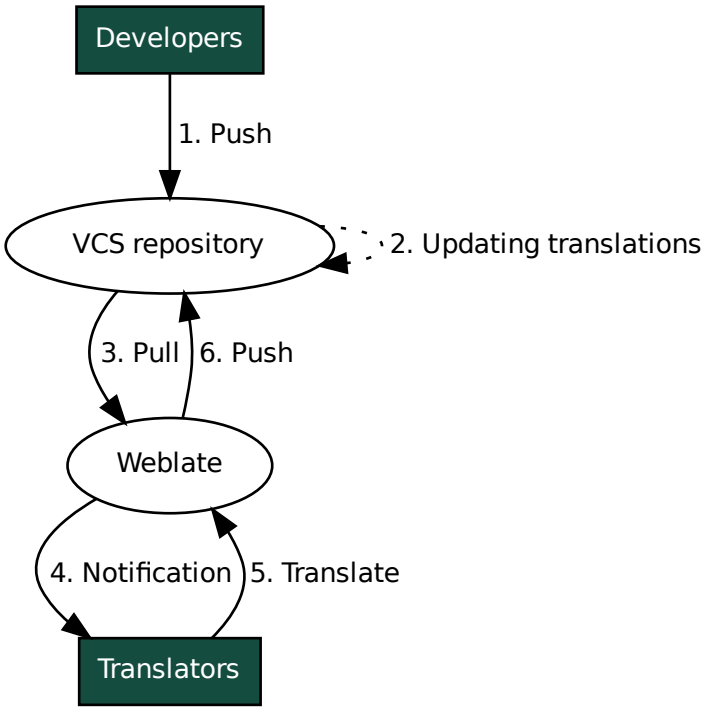
Veja também:

/devel/integration descreve maneiras básicas de integrar o seu desenvolvimento com o Weblate.

O processo é o seguinte:

1. Os programadores fazem alterações e fazem *push* delas so repositório VCS.
2. Opcionalmente, os ficheiros de tradução são atualizados (isso depende do formato do ficheiro, consulte *Por que é que o Weblate ainda mostra as cadeias de tradução antigas quando atualizei o modelo?*).
3. O Weblate faz o *pull* das alterações do repositório VCS, consulte *Atualizar repositórios*.
4. Uma vez que o Weblate deteta alterações nas traduções, os tradutores são notificados com base na configurações de assinatura deles.
5. Os tradutores enviam traduções a usar a interface web do Weblate ou enviam alterações feitas offline.

6. Uma vez que os tradutores acabaram, o Weblate faz commit das alterações no repositório local (veja *Commits adiados*) e faz *push* delas de volta se tiver permissões para fazê-lo (veja *Fazendo push das alterações do Weblate*).



2.9.1 Atualizar repositórios

Deve configurar alguma maneira de atualizar repositórios de backend a partir da fonte dele.

- Use *Hooks de notificação* para integrar com a maioria dos serviços comuns de hospedagem de código:
  - *Receber alterações do GitHub automaticamente*
  - *Receber alterações do GitLab automaticamente*
  - *Receber alterações do Bitbucket automaticamente*
  - *Receber alterações do Pagure automaticamente*
  - *Receber alterações dos Azure Repos automaticamente*
- Acione manualmente a atualização na gestão do repositório ou a usar *API REST do Weblate* ou *Cliente Weblate*
- Ative `AUTO_UPDATE` para atualizar todos os componentes na sua instância Weblate automaticamente
- Execute `updategit` (com a seleção de um projeto ou `--all` para atualizar tudo)

Sempre que o Weblate atualiza o repositório, as extensões de pós-atualização serão acionadas, consulte *Extensões*.

Evitar conflitos de mesclagem

Os conflitos de mesclagem do Weblate surgem quando o mesmo ficheiro foi alterado tanto no Weblate quanto fora dele. Existem duas abordagens para lidar com isso - evitar edições fora do Weblate ou integrar o Weblate no seu processo de atualização, de modo que descarte alterações antes de atualizar os ficheiros fora do Weblate.

A primeira abordagem é fácil com ficheiros monolingues - pode adicionar novas cadeias no Weblate e deixar a edição completa dos ficheiros lá. Para ficheiros bilíngues, geralmente há algum tipo de processo de extração de mensagens para gerar ficheiros traduzíveis do código-fonte. Em alguns casos, isso pode ser dividido em duas partes - uma para a extração gera modelo (por exemplo, o GETTEXT POT é gerado a usar **`xgettext`**) e depois o processo a seguir mescla-o em traduções reais (os ficheiros GETTEXT PO são atualizados a usar **`msgmerge`**). Pode executar o segundo passo dentro do Weblate e garantirá que todas as alterações pendentes sejam incluídas antes desta operação.

A segunda abordagem pode ser alcançada a usar o *API REST do Weblate* para forçar o Weblate a fazer push de todas as alterações pendentes e bloquear a tradução enquanto está fazendo alterações do seu lado.

O script para fazer atualizações pode ser assim:

```
# Lock Weblate translation
wlc lock
# Push changes from Weblate to upstream repository
wlc push
# Pull changes from upstream repository to your local copy
git pull
# Update translation files, this example is for Django
./manage.py makemessages --keep-pot -a
git commit -m 'Locale updates' -- locale
# Push changes to upstream repository
git push
# Tell Weblate to pull changes (not needed if Weblate follows your repo
# automatically)
wlc pull
# Unlock translations
wlc unlock
```

Se tiver vários componentes compartilhando o mesmo repositório, deve bloqueá-los todos separadamente:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Nota: O exemplo usa *Cliente Weblate*, que precisa de configuração (chaves de API) para ser capaz de controlar o Weblate remotamente. Também pode conseguir isso a usar qualquer cliente HTTP em vez de wlc, por exemplo, curl, ver *API REST do Weblate*.

Veja também:

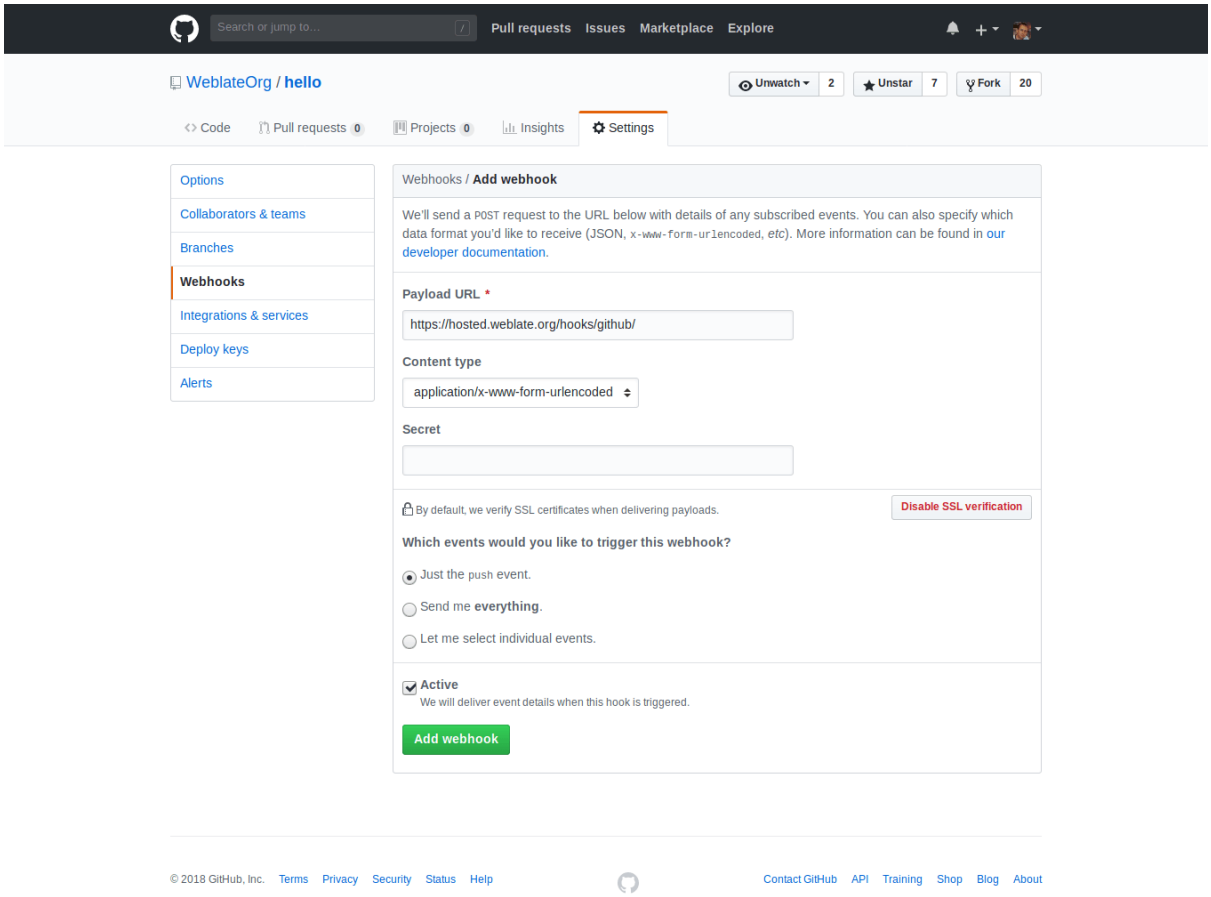
Cliente Weblate

Receber alterações do GitHub automaticamente

O Weblate vem com suporte nativo ao GitHub.

Se estiver a usar o Hosted Weblate, a abordagem recomendada é instalar o app Weblate, dessa forma terá a configuração correta sem ter que configurar muito. Também pode ser usado para fazer push de mudanças de volta.

Para receber notificações em cada push a um repositório do GitHub, adicione o webhook do Weblate nas configurações do repositório (*Webhooks*) como mostrado na imagem abaixo:



Para a URL de carga útil, anexar `/hooks/github/` à URL do Weblate, por exemplo, para o serviço Hosted Weblate, este é `https://hosted.weblate.org/hooks/github/`.

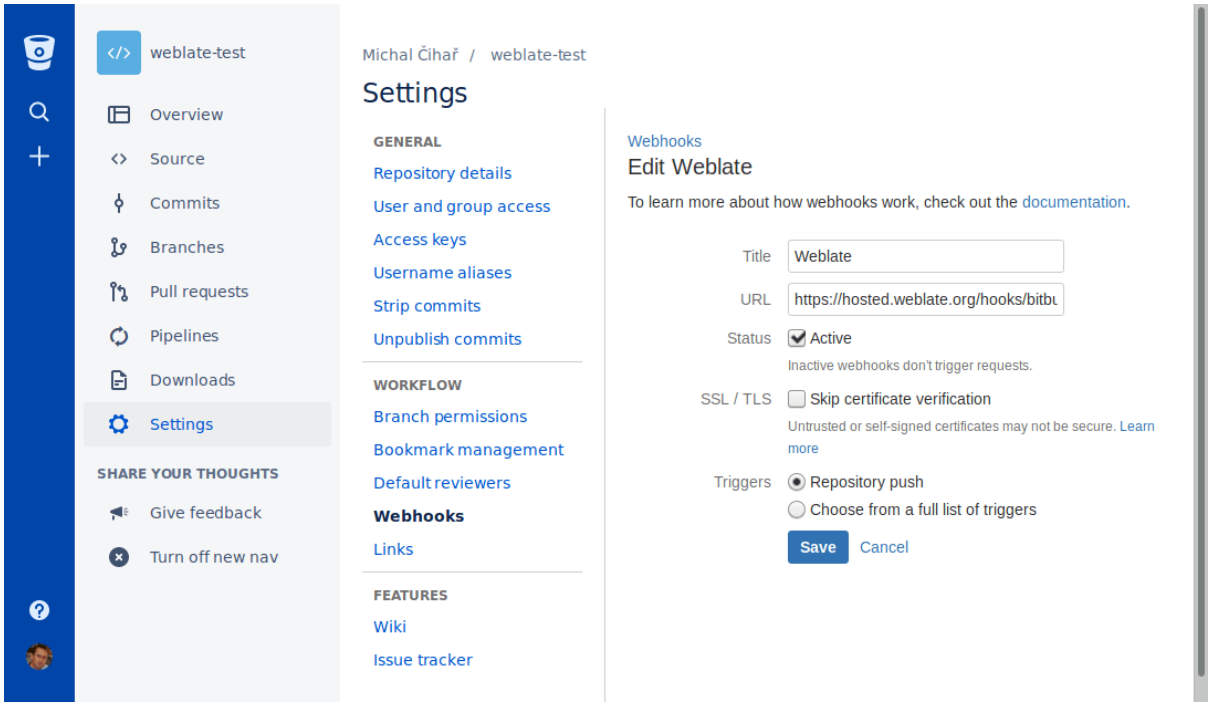
Pode deixar outros valores nas configurações predefinidas (o Weblate pode lidar com ambos os tipos de conteúdo e consome apenas o evento *push*).

Veja também:

`POST /hooks/github/`, *Acessando repositórios do Hosted Weblate*

Receber alterações do Bitbucket automaticamente

O Weblate tem suporte para webhooks do Bitbucket, adicione um webhook que aciona no push do repositório, com destino a URL `/hooks/bitbucket/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/bitbucket/`).



Veja também:

`POST /hooks/bitbucket/`, *Acessando repositórios do Hosted Weblate*

Receber alterações do GitLab automaticamente

O Weblate tem suporte para ganchos do GitLab, adiciona um webhook de projeto com destino a URL `/hooks/gitlab/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitlab/`).

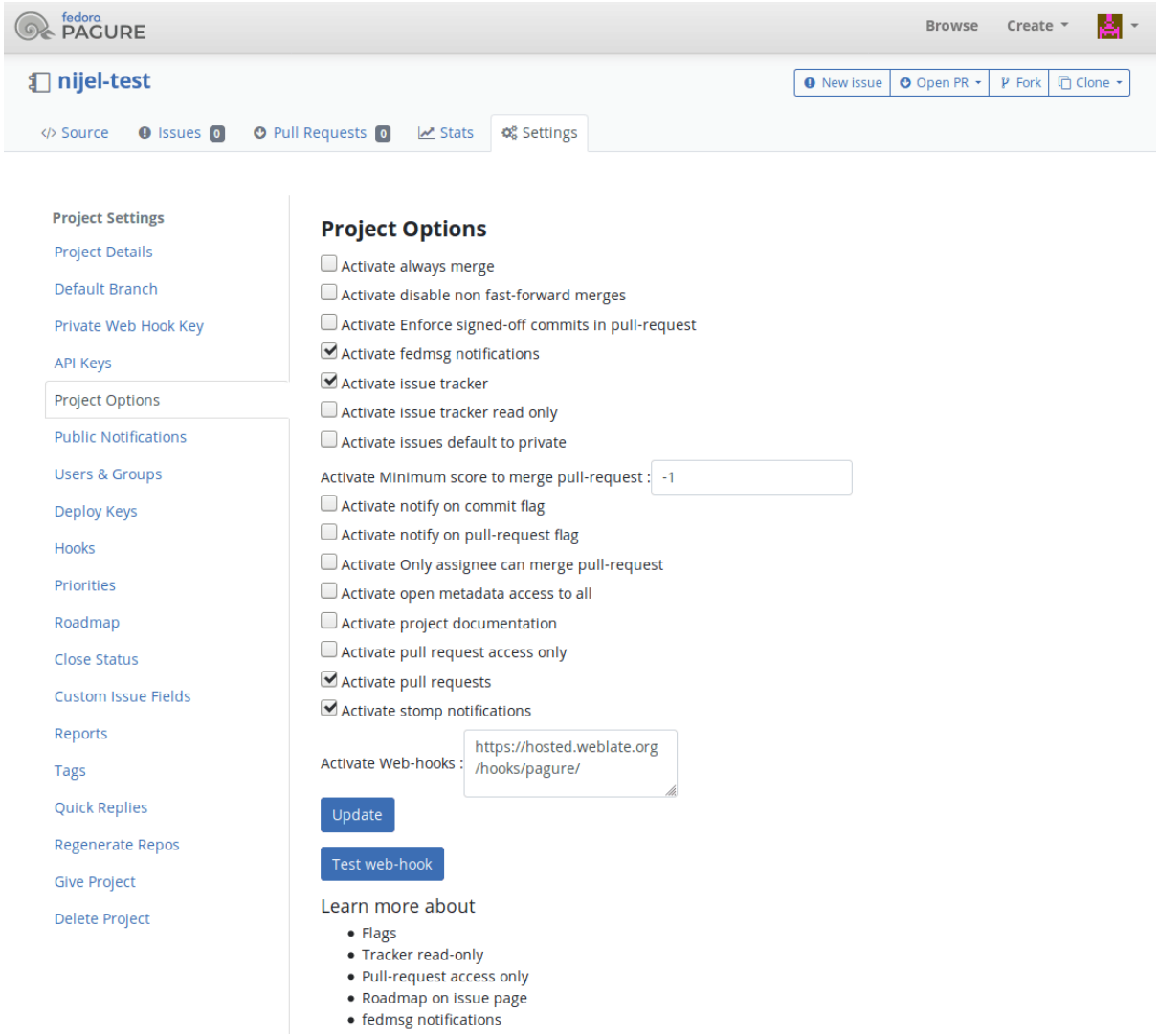
Veja também:

`POST /hooks/gitlab/`, *Acessando repositórios do Hosted Weblate*

Receber alterações do Pagure automaticamente

Novo na versão 3.3.

O Weblate tem suporte para ganchos Pagure. Adicione um webhook com destino a URL `/hooks/pagure/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/pagure/`). Isso pode ser feito em *Web-hooks* em *Project options*:



Veja também:

`POST /hooks/pagure/`, *Acessando repositórios do Hosted Weblate*

Receber alterações dos Azure Repos automaticamente

Novo na versão 3.8.

O Weblate tem suporte para webhooks dos Azure Repos, adicione um webhook para o evento *Code pushed* com destino para URL `/ganchos/azure/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/azure/`). Isso pode ser feito em *Service hooks `em :guilabel:`Project settings*.

Veja também:

Webhooks no manual do Azure DevOps, `POST /hooks/azure/`, *Acessando repositórios do Hosted Weblate*

Receber alterações dos Gitea Repos automaticamente

Novo na versão 3.9.

Weblate tem suporte para webhooks do Gitea, adicione um *Gitea Webhook* para o evento *Push events* com destino ao URL `/hooks/gitea/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitea/`). Isso pode ser feito no *Webhooks* em *Settings* do repositório.

Veja também:

Webhooks no manual do Gitea, `POST /hooks/gitea/`, *Acessando repositórios do Hosted Weblate*

Receber alterações de Gitee Repos automaticamente

Novo na versão 3.9.

O Weblate tem suporte para webhooks Gitee, adicione um *WebHook* para o evento *Push* com destino para URL `/hooks/gitee/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitee/`). Isso pode ser feito em *WebHooks* sob *Management* do repositório.

Veja também:

Webhooks no manual do Gitee, `POST /hooks/gitee/`, *Acessando repositórios do Hosted Weblate*

Atualizar repositórios *nightly* automaticamente

O Weblate busca automaticamente repositórios remotos *nightly* para melhorar o desempenho ao mesclar alterações mais tarde. Pode opcionalmente transformar isso em fazer mesclagens noturnas também, ativando `AUTO_UPDATE`.

2.9.2 Fazendo push das alterações do Weblate

Cada componente de tradução pode ter uma URL de push configurada (veja *URL de submissão do repositório*) e, nesse caso, o Weblate será capaz de fazer push da alteração ao repositório remoto. O Weblate também pode ser configurado para fazer push automaticamente das alterações em cada commit (isso é a predefinição, veja *Enviar ao submeter*). Se não quiser que seja feito push automático das alterações, pode fazer-lo manualmente em *Manutenção do repositório* ou a usar API via `wlc push`.

As opções de push diferem com base no *Integração de controlo de versões* usado, mais detalhes são encontrados nesse capítulo.

No caso de não querer pushes diretos pelo Weblate, há suporte para pull requests do *GitHub pull requests*, do *Solicitação de mesclagem do GitLab* e do *Pagure merge requests* ou revisões do *Gerrit*, pode ativá-los a escolher *GitHub*, *GitLab*, *Gerrit* ou *Pagure*, como *Sistema de controlo de versões* em *Configuração de componente*.

No geral, as opções a seguir estão disponíveis com Git, GitHub e GitLab:

Configuração desejada	*Sistema de controlo de versões*	*URL de submissão do repositório*	*Ramo do push*
Sem push	*Git*	*vazio*	*vazio*
Push diretamente	*Git*	URL de SSH	*vazio*
Empurrar para um ramo separado	*Git*	URL de SSH	Nome do ramo
Pull request de GitHub do fork	*GitHub pull requests*	*vazio*	*vazio*
Pull request de GitHub do ramo	*GitHub pull requests*	URL de SSH[1]	Nome do ramo
Merge request de GitLab do fork	*Solicitação de mesclagem do GitLab*	*vazio*	*vazio*
Merge request de GitLab do ramo	*Solicitação de mesclagem do GitLab*	URL de SSH[Página 290, 1]	Nome do ramo
Merge request de Pagure do fork	*Pagure merge requests*	*vazio*	*vazio*
Merge request de Pagure do ramo	*Pagure merge requests*	URL de SSH[1]	Nome do ramo

Nota: Também pode ativar o push automático de alterações após o Weblate fazer commit, isso pode ser feito em *Enviar ao submeter*.

Veja também:

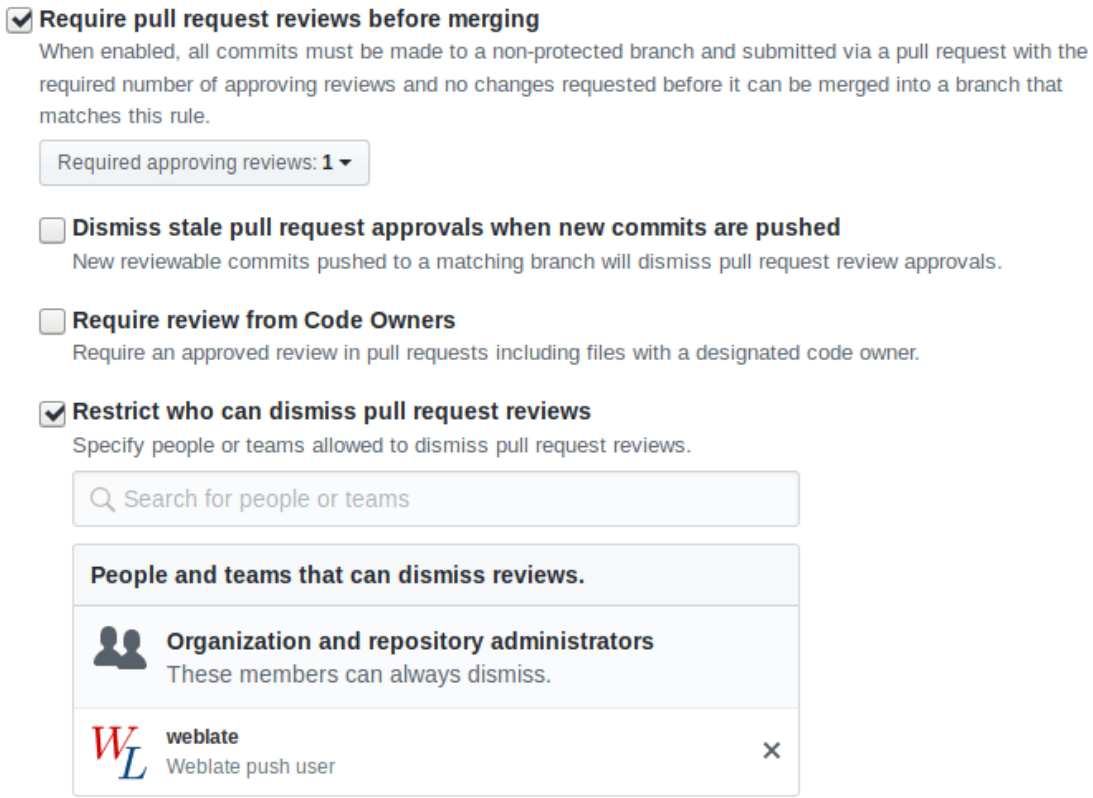
Consulte *Acessando repositórios* para configurar chaves de SSH e *Commits adiados* para obter informações sobre quando o Weblate decide fazer commit de alterações.

Ramos protegidos

Se estiver a usar o Weblate em ramo protegido, pode configurá-lo para usar pull requests e executar revisão real sobre as traduções (o que pode ser problemático para idiomas que não conhece). Uma abordagem alternativa é abrir mão desta limitação em favor do utilizador de push no Weblate.

Por exemplo, no GitHub, isso pode ser feito na configuração do repositório:

[1] Pode estar vazia caso o *Repositório do código-fonte* tenha suporte a push.



2.9.3 Interagir com os outros

O Weblate facilita a interação com outras pessoas a usar a API dele.

Veja também:

API REST do Weblate

2.9.4 Commits adiados

O comportamento do Weblate é de agrupar commits do mesmo autor num só commit, se for possível. Isso reduz a quantidade de commits consideravelmente, no entanto, pode precisar de dizer explicitamente para fazer os commits no caso de querer deixar o repositório VCS em sincronia, por exemplo, para mesclarem (isso é por predefinição permitido para o grupo *Managers*, consulte *Lista de privilégios e funções internas*).

As alterações neste modo têm o commit delas feitas assim que qualquer uma das seguintes condições são cumpridas:

- Outra pessoa altera uma cadeia já alterada.
- Um merge do upstream é feito.
- Um commit explícito é solicitado.
- A file download is requested.
- A alteração é mais antiga do que o período definido como *Idade das alterações a fazer commit* em *Configuração de componente*.

Dica: Os commits são criados para cada componente. Então, caso tenha muitos componentes, ainda verá muitos compromissos. Pode utilizar a extensão *Squash de commits git* neste caso.

Se quiser fazer commit das alterações com mais frequência e sem verificar a idade, pode agendar uma tarefa regular para realizar um commit:

```
CELERY_BEAT_SCHEDULE = {
    # Unconditionally commit all changes every 2 minutes
    "commit": {
        "task": "weblate.trans.tasks.commit_pending",
        # Omitting hours will honor per component settings,
        # otherwise components with no changes older than this
        # won't be committed
        "kwargs": {"hours": 0},
        # How frequently to execute the job in seconds
        "schedule": 120,
    }
}
```

2.9.5 Processar repositório com scripts

A maneira de personalizar como o Weblate interage com o repositório é com *Extensões*. Consulte *Escrevendo scripts para extensões* para obter informações sobre como executar scripts externos através de extensões.

2.9.6 Manter traduções iguais entre componentes

Uma vez que tenha vários componentes de tradução, pode garantir que as mesmas cadeias tenham a mesma tradução. Isso pode ser alcançado em vários níveis.

Propagação de tradução

Com *Permitir propagação da tradução* ativada (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 utilizador que traduz atualmente em todos os componentes.

Nota: A propagação de tradução requer a chave para ser compatível com formatos de tradução monolíngue, por isso tenha isso em mente ao criar chaves de tradução.

Verificação de consistência

A verificação check-inconsistente é acionada sempre que as cadeias são diferentes. Pode usar isso para rever tais diferenças manualmente e escolher a tradução certa.

Tradução automática

A tradução automática com base em diferentes componentes pode ser uma maneira de sincronizar as traduções entre os componentes. Pode acioná-la manualmente (veja *Tradução automática*) ou fazê-la ser executada automaticamente na atualização do repositório usando uma extensão (veja *Tradução automática*).

2.10 Licenciar traduções

Pode especificar sob quais traduções de licença são contribuídas. Isto é especialmente importante se as traduções forem abertas ao público, para estipular para que elas possam ser usadas.

Deve especificar as informações da licença da *Configuração de componente*. Deve evitar exigir um contrato de licença de colaborador, embora seja possível.

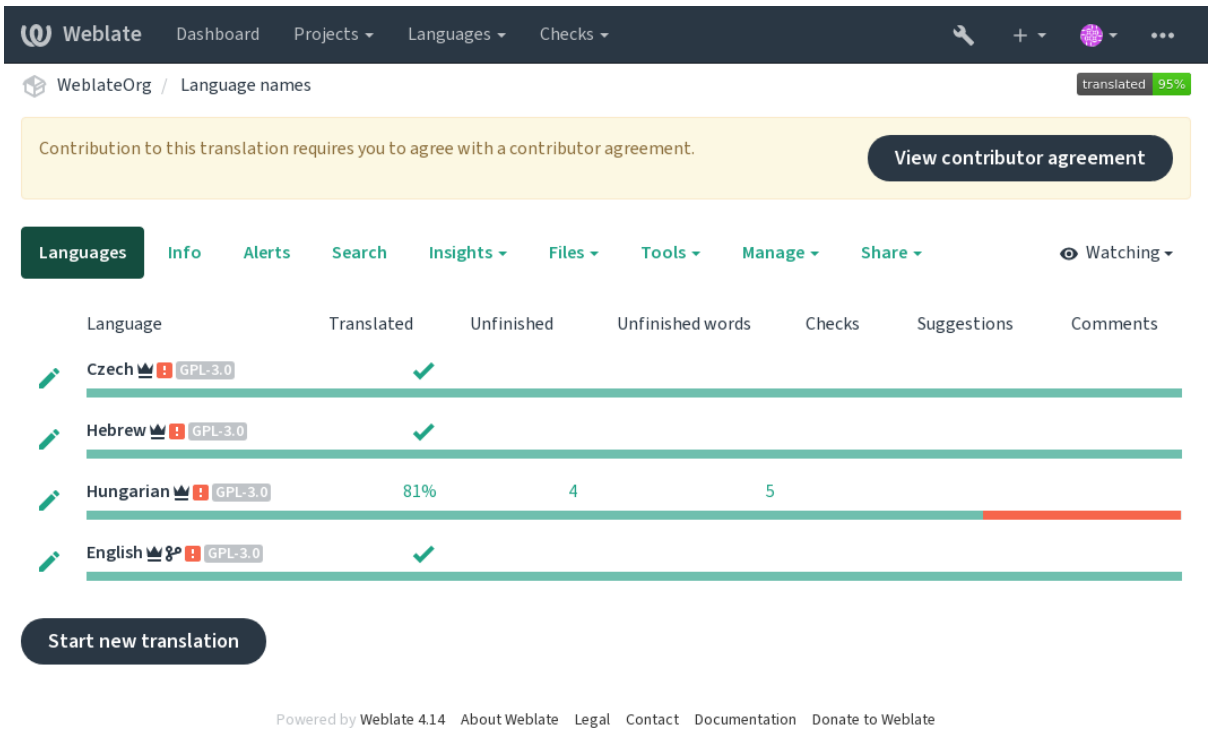
2.10.1 Informações de licença

Ao especificar informações de licenças (nome da licença e URL), essas informações são mostradas na secção de informações de tradução do respetivo *Configuração de componente*.

Normalmente este é o melhor lugar para publicar informações de licenciamento se nenhum consentimento explícito for necessário. Se o seu projeto ou tradução não for livre, provavelmente precisa de consentimento prévio.

2.10.2 Acordo de contribuidor

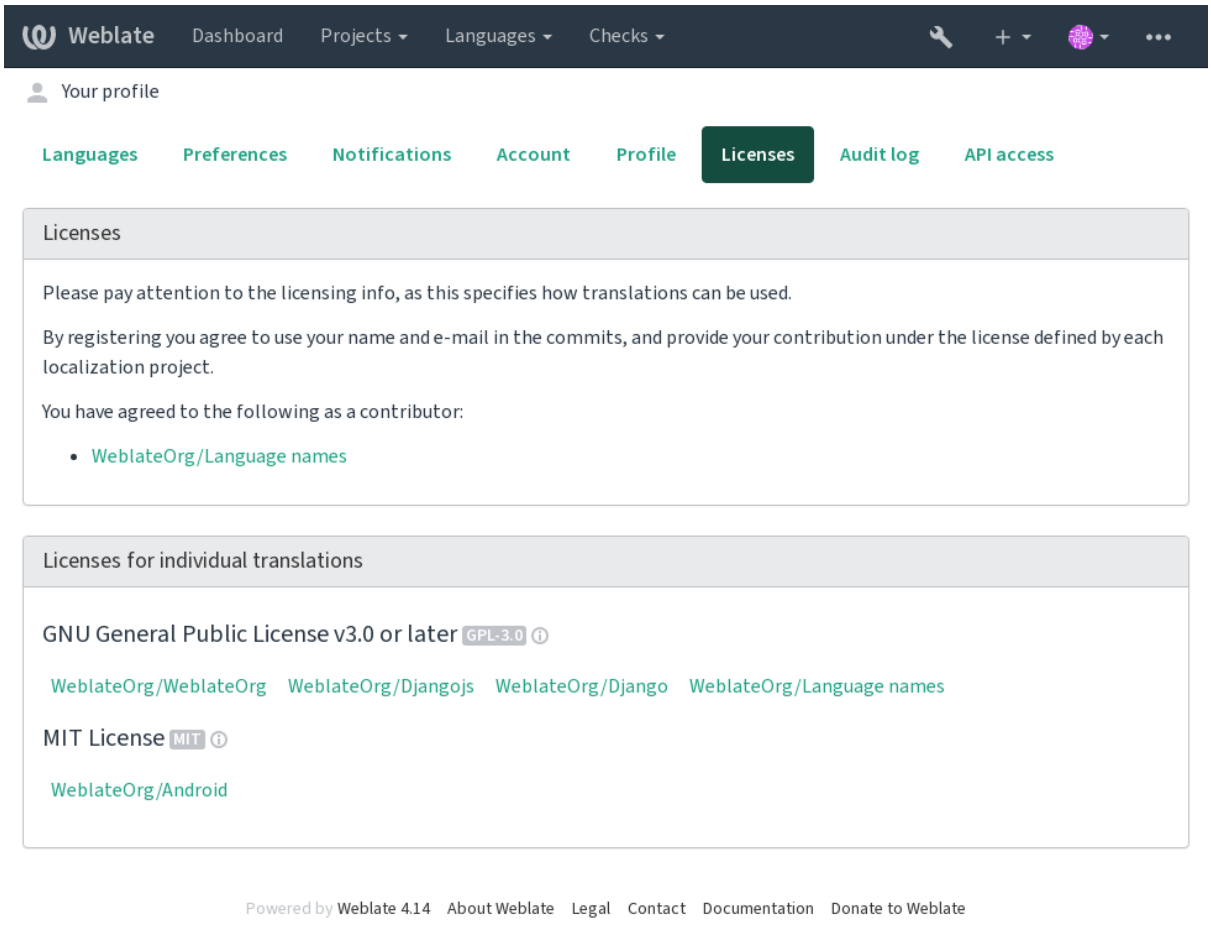
Se especificar um contrato de licença de colaborador, apenas os utilizadores que concordaram com ele poderão contribuir. Este é um passo claramente visível ao acessar a tradução:



O texto inserido é formatado em parágrafos e ligações externas podem ser incluídas. A marcação HTML não pode ser usada.

2.10.3 Licenças de utilizador

Todos utilizadores podem rever todas as licenças de tradução de todos os projetos públicos na instância do seu perfil:



2.11 Processo de tradução

2.11.1 Votação de sugestão

Todos podem adicionar sugestões por padrão, para serem aceitas por utilizadores autenticados. A votação de sugestões pode ser usada para fazer o uso de um texto quando mais de um utilizador autenticado concorda, ao configurar a *Configuração de componente* com *Votação de sugestões* para ativar a votação, e *Aceitar sugestões automaticamente* para estipular um limite para sugestões aceitas (isso inclui um voto do utilizador a fazer a sugestão, se ela for lançada).

Nota: Uma vez que a aceitação automática está configurada, utilizadores normais perdem o privilégio de gravar diretamente traduções ou aceitar sugestões. Isso pode ser sobrescrito pela *permissão Editar texto quando as sugestões são forçadas*.

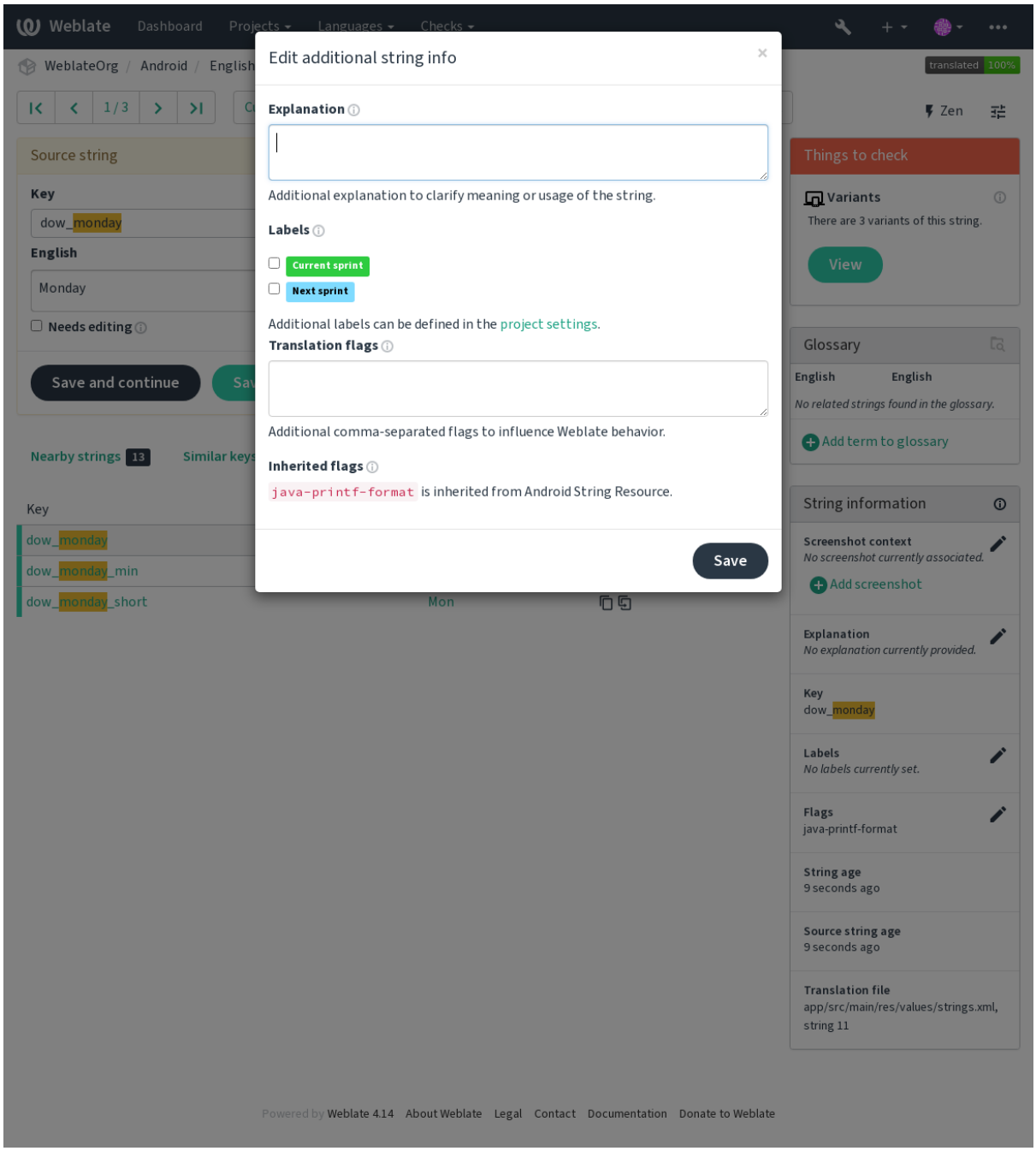
Pode combiná-las com :ref:`controle de acesso <access-control>`para uma das configurações a seguir:

- Utilizadores sugerem e votam por sugestões e um grupo limitado controla o que é aceito. - Ativar votação. - Desativar aceitação automática. - Não deixar utilizadores gravarem traduções.
- Utilizadores sugerem e votam em sugestões com aceitação automática uma vez que a quantidade definida deles concorda. - Ativar votação. - Definir o numero desejado de votos para aceitação automática.

- Voto opcional para sugestões. (Pode ser usado opcionalmente pelos utilizadores quando não têm certeza sobre uma tradução, marcando múltiplas sugestões.) - Apenas ativar a votação.

2.11.2 Informações adicionais sobre cadeias fonte

Melhore o processo de tradução adicionando informações extra para os textos incluindo explicações, prioridades dos textos, marcadores de verificações e contexto visual. Algumas dessas informações podem ser extraídas dos ficheiros de tradução e podem ser adicionadas ao editar as informações adicionais do texto:



Acesse isso diretamente da interface de tradução clicando no ícone «Editar» perto de *Contexto da captura de ecrã* ou *Marcadores*.

Weblate Dashboard Projects Languages Checks

WeblateOrg / Django / Czech / Translate translated 96%

11 / 26 All strings Position and priority Zen

Translation

Explanation

Help text for automatic translation tool

English

Automatic translation via machine translation uses active machine translation engines to get the best possible translations and applies them in this project.

Czech

Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt.

Needs editing 169/1570 · 157

Save and continue | Save and stay | Suggest | Skip

Nearby strings 26 | Comments | Automatic suggestions | Other languages 4 | History

Context	English	Czech	Actions
	Files	Soubory	
	Automatic translation	Automatický překlad	
	Add new translation string	Add new translation string	
	Translation status	Stav překladu	
	%(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.	Nahraný soubor bude sloučen se stávajícími překlady. Pokud chcete přepsat již přeložené řetězce, nezapomeňte to povolit.	
	The uploaded file will be merged with the current translation.	Nahraný soubor bude sloučen se stávajícími překlady.	
	The fulltext search might not work properly as the fulltext index for this translation is not yet up to date.	Fulltextové vyhledávání nemusí fungovat správě, protože fulltextový index pro tento překlad ještě není plně zpracován.	
	Review	Kontrola	
	Review translations touched by other users.	Zkontrolovat překlady od ostatních uživatelů.	
	Start review	Začít kontrolu	
	Percent	Procenta	
	Total	Celkem	
	Failing check	Neúspěšných kontrol	
	Last activity	Poslední aktivita	
	Last change	Poslední změna	
	Last author	Poslední autor	
Question for a mathematics-based CAPTCHA, the %s is an arithmetic problem	What is %s?	Kolik to je?	
	The string uses three dots (...) instead of an ellipsis character (…)		

Glossary

English	Czech	
machine translation	strojový překlad	WeblateOrg
project	projekt	WeblateOrg

Add term to glossary

String information

Screenshot context
No screenshot currently associated.
Add screenshot

Explanation
Help text for automatic translation tool

Labels
No labels currently set.

Flags
No flags currently set.

Source string location
weblate/templates/translation.html:212

String age
a second ago

Source string age
2 seconds ago

Translation file
weblate/locale/cs/LC_MESSAGES/djangojs.po, string 11

Powered by Weblate 4.14 About Weblate Legal Contact Documentation Donate to Weblate

Priorização dos textos

Novo na versão 2.0.

A prioridade dos textos pode ser mudada para oferecer textos prioritários para tradução antes, usando o marcador `priority`.

Dica: Isso pode ser utilizado para organizar a ordem de tradução de maneira lógica.

Veja também:

Verificações de qualidade

Marcadores de tradução

Novo na versão 2.4.

Alterado na versão 3.3: Anteriormente chamado de *Marcadores de verificações de qualidade*, ele já não configura apenas verificações.

Personalização de verificações de qualidade e outros comportamentos do Weblate, veja *Personalizando comportamento a usar marcadores*.

Os marcadores de textos também são herdados dos *Marcadores de tradução* na *Configuração de componente* e marcadores do ficheiro de tradução (veja *Formatos de ficheiros suportados*).

Veja também:

Verificações de qualidade, *Personalizando comportamento a usar marcadores*

Explicação

Alterado na versão 4.1: Nas versões anteriores, isso era chamado de *Contexto extra*.

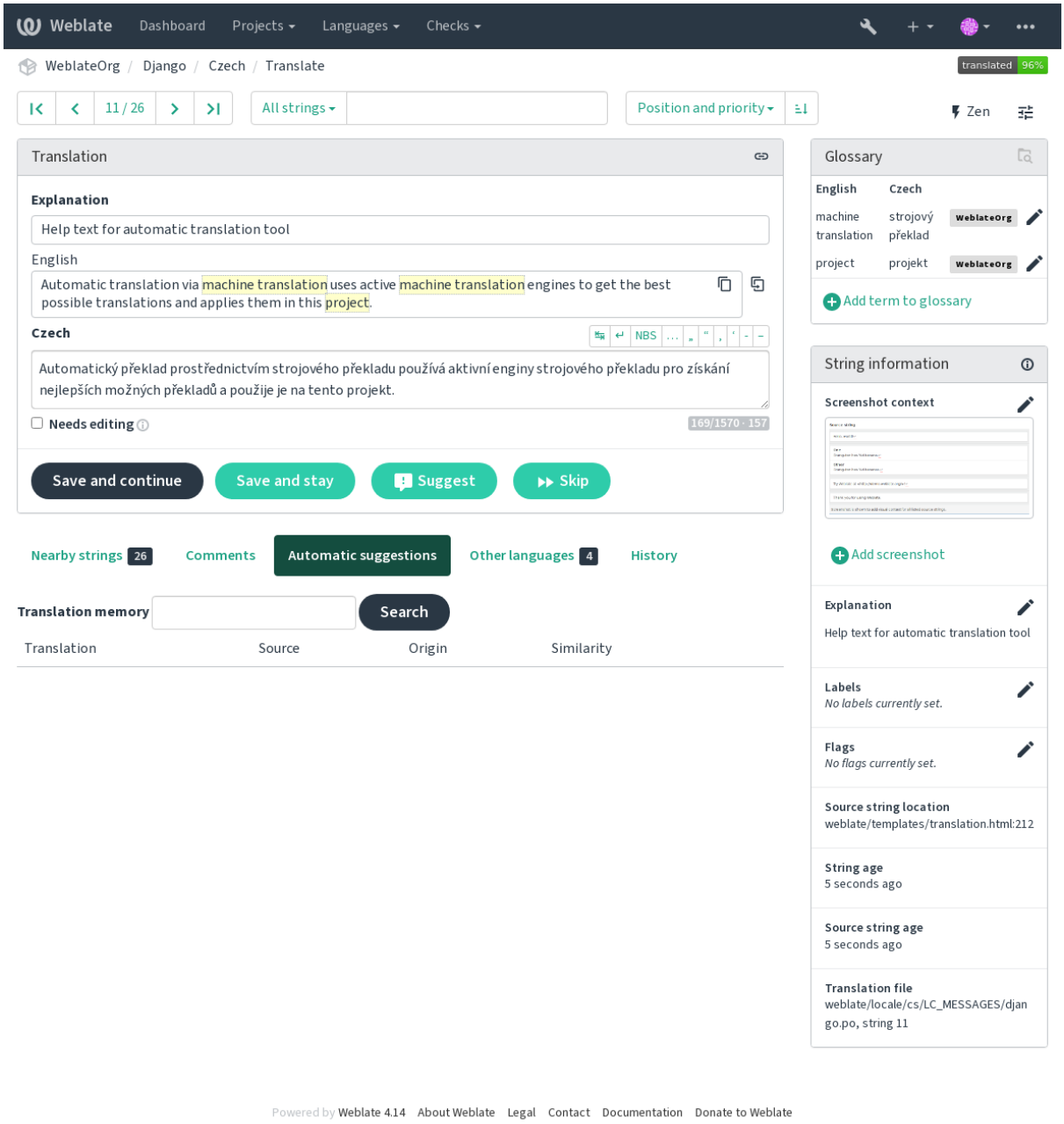
Use a explicação para esclarecer o escopo ou uso da tradução. Pode usar Markdown para incluir ligações e outras marcações.

Contexto visual para cadeias

Novo na versão 2.9.

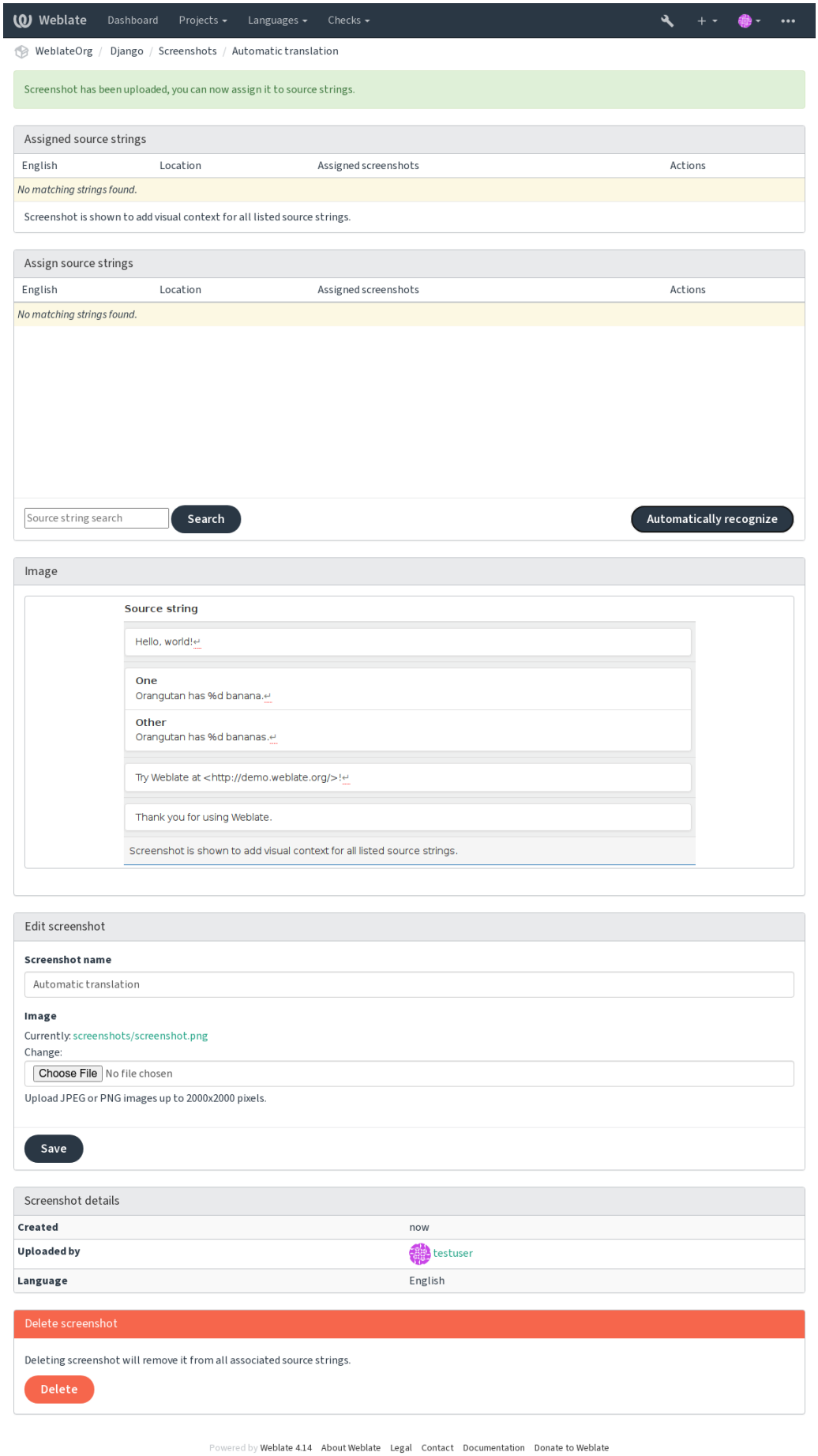
Pode enviar uma captura de ecrã que mostram um determinado texto fonte em uso no seu programa. Isso ajuda os tradutores a entender onde é usado e como deve ser traduzido.

A captura de ecrã enviada é mostrada na barra lateral de contexto de tradução:



Além de *Informações adicionais sobre cadeias fonte*, as capturas de ecrã têm uma interface de gestão separada no menu *Ferramentas*. Envie capturas de ecrã, atribua-as a textos fonte manualmente ou use o reconhecimento ótico de caracteres para fazer isso.

Depois que uma captura de ecrã é carregada, esta interface lida com a gestão e a associação de texto fonte:


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Django / Screenshots / Automatic translation
Screenshot has been uploaded, you can now assign it to source strings.
Assigned source strings
English
Location
Assigned screenshots
Actions
No matching strings found.
Screenshot is shown to add visual context for all listed source strings.
Assign source strings
English
Location
Assigned screenshots
Actions
No matching strings found.
Source string search
Search
Automatically recognize
Image
Source string
Hello, world!
One
Orangutan has %d banana.
Other
Orangutan has %d bananas.
Try Weblate at <http://demo.weblate.org/>!
Thank you for using Weblate.
Screenshot is shown to add visual context for all listed source strings.
Edit screenshot
Screenshot name
Automatic translation
Image
Currently: screenshots/screenshot.png
Change:
Choose File
No file chosen
Upload JPEG or PNG images up to 2000x2000 pixels.
Save
Screenshot details
Created
now
Uploaded by
testuser
Language
English
Delete screenshot
Deleting screenshot will remove it from all associated source strings.
Delete
Powered by Weblate 4.14
About Weblate
Legal
Contact
Documentation
Donate to Weblate

2.12 Verificações e correções

2.12.1 Correções automáticas personalizadas

Também pode implementar a sua própria correção automática, além das predefinidas e incluí-las em *AUTOFIX_LIST*.

As correções automáticas são poderosas, mas também podem causar danos; tenha cuidado ao escrever uma.

Por exemplo, a correção automática a seguir iria substituir cada ocorrência da cadeia `foo`, numa tradução com `bar`:

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#

from django.utils.translation import gettext_lazy as _

from weblate.trans.autofixes.base import AutoFix


class ReplaceFooWithBar(AutoFix):
    """Replace foo with bar."""

    name = _("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False
```

Para instalar verificações personalizadas, forneça um caminho totalmente qualificado à classe Python em *AUTOFIX_LIST*, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

2.12.2 Personalizando comportamento a usar marcadores

You can fine-tune the Weblate behavior by using flags. This can be done on the source string level (see *Informações adicionais sobre cadeias fonte*), or in the *Configuração de componente* (*Marcadores de tradução*). Some file formats also allow to specify flags directly in the format (see *Formatos de ficheiros suportados*).

As etiquetas são separadas por vírgulas, os parâmetros são separados por caracteres de dois pontos. Pode usar aspas para incluir espaços em branco ou caracteres especiais na cadeia. Por exemplo:

```
placeholders:"special:value":"other value", regex:.*
```

Aspas simples e duplas são aceitas e caracteres especiais são tratados com barra invertida:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

Aqui está uma lista de sinalizadores atualmente aceitos:

rst-text
Trata um texto como um documento reStructuredText, afeta *Tradução inalterada*.

dos-eol
Usa marcadores de ponta de linha do DOS em vez dos Unix (`\r\n` em vez de `\n`).

read-only
A cadeia é somente leitura e não deve ser editada no Weblate, veja *Textos somente leitura*.

priority:N
Prioridade da cadeia. As cadeias de maior prioridade são apresentados primeiro para tradução. A prioridade predefinida é 100, quanto maior prioridade que um texto tem, mais cedo é oferecido para tradução.

max-length:N
Limita o comprimento máximo de uma cadeia a N caracteres, veja *Tamanho máximo da tradução*.

xml-text
Trata o texto como documento XML, afeta *Sintaxe XML* e *Markup XML*.

font-family:NOME
Define a família de letras para verificações da renderização, veja *Gerir letras*.

font-weight:PESO
Define o peso da letra para verificações da renderização, veja *Gerir letras*.

font-size:SIZE
Define o tamanho da letra para verificações da renderização, veja *Gerir letras*.

font-spacing:ESPAÇAMENTO
Define o espaçamento de letras para verificações da renderização, veja *Gerir letras*.

icu-flags:MARCADORES
Define marcadores para personalizar o comportamento da verificação de qualidade *MessageFormat do ICU*.

icu-tag-prefix:PREFIXO
Define um prefixo necessário para tags XML para a verificação de qualidade *MessageFormat do ICU*.

placeholders:NOME:NOME2:...
Cadeias de espaço reservado esperados na tradução, veja *Espaços reservados*.

replacements:DE:PARA:DE2:PARA2...
Substituições para realizar ao verificar parâmetros de texto resultantes (por exemplo, em *Tamanho máximo da tradução* ou *Tamanho máximo da tradução*). O caso de uso típico para isso é expandir objetos colocáveis *(placeables)* para garantir que o texto se encaixe mesmo com valores longos, por exemplo: `replacements:%s:"John Doe"`.

variants:FONTE
Marca esta cadeia como uma variante do texto com a fonte correspondente. Veja variants.

regex:EXPRESSÃO REGULAR
Expressão regular para corresponder à tradução; veja *Expressão regular*.

forbidden
Indica tradução proibida num glossário, veja *Traduções proibidas*.

strict-same
Faz com que «Tradução não alterada» evite usar a lista negra de palavras embutidas, veja *Tradução inalterada*.

check-glossary
Ative a verificação de qualidade *Não segue o glossário*.

angularjs-format
Ative a verificação de qualidade *Cadeia de interpolação AngularJS*.

c-format
Ative a verificação de qualidade *Formato C*.

c-sharp-format
Ative a verificação de qualidade *Formato C#*.

es-format
Ative a verificação de qualidade :ref: *check-es-format*.

i18next-interpolation
Ative a verificação de qualidade *Interpolação de i18next*.

icu-message-format
Ativa a verificação de qualidade *MessageFormat do ICU*.

java-printf-format
Enable the *Formato Java* quality check.

java-format
Ative a verificação de qualidade *Formato de Mensagem Java*.

javascript-format
Ative a verificação de qualidade :ref:"check-javascript-format".

lua-format
Ative a verificação de qualidade *Formato Lua*.

object-pascal-format
Ative a verificação de qualidade *Formato Object Pascal*.

percent-placeholders
Ativa a verificação de qualidade *Espaços reservados de percentagem*.

perl-format
Ativa a verificação de qualidade *Formato Perl*.

php-format
Ativa a verificação de qualidade *Formato PHP*.

python-brace-format
Ativa a verificação de qualidade *Formato de chaveta Python*.

python-format
Ativa a verificação de qualidade *Formato Python*.

qt-format
Ativa a verificação de qualidade *Formato Qt*.

qt-plural-format
Ativa a verificação de qualidade *Forma plural Qt*.

ruby-format
Ativa a verificação de qualidade *Formato Ruby*.

scheme-format
Ativa a verificação de qualidade *Formato Scheme*.

vue-format
Ativa a verificação de qualidade *Formatação vue I18n*.

md-text
Trata o texto como um documento Markdown. Ativa as verificações de qualidade afeta *Hiperligações de marcação*, *Referências de Markdown* e *Sintaxe de Markdown*.

case-insensitive
Adjust checks behavior to be case-insensitive. Currently affects only *Espaços reservados* quality check.

safe-html
Ativa a verificação de qualidade *HTML inseguro*.

url
O texto deve consistir apenas numa URL. Ativa a verificação de qualidade *URL*.

ignore-all-checks
Ignore todas as verificações de qualidade.

ignore-bbcode
Ignora a verificação de qualidade *Markup BBCode*.

ignore-duplicate
Ignora a verificação de qualidade *Palavras consecutivas duplicadas*.

ignore-check-glossary
Ignora a verificação de qualidade *Não segue o glossário*.

ignore-double-space
Ignora a verificação de qualidade *Espaço duplo*.

ignore-angularjs-format
Ignora a verificação de qualidade *Cadeia 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-icu-message-format
Ignora a verificação de qualidade *MessageFormat do ICU*.

ignore-java-format
Ignora a verificação de qualidade *Formato de Mensagem Java*.

ignore-java-printf-format
Skip the *Formato Java* quality check.

ignore-javascript-format
Ignora a verificação de qualidade *Formato JavaScript*.

ignore-lua-format
Ignora a verificação de qualidade *Formato Lua*.

ignore-object-pascal-format
Ignora a verificação de qualidade *Formato Object Pascal*.

ignore-percent-placeholders
Ignora a verificação de qualidade *Espaços reservados de percentagem*.

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 chaveta 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 *Forma plural Qt*.

ignore-ruby-format
Ignora a verificação de qualidade *Formato Ruby*.

ignore-scheme-format
Ignora a verificação de qualidade *Formato Scheme*.

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 utilizada*.

ignore-md-link
Ignora a verificação de qualidade *Hiperligações de marcação*.

ignore-md-reflink
Ignora a verificação de qualidade *Referências de Markdown*.

ignore-md-syntax
Ignora a verificação de qualidade *Sintaxe de Markdown*.

ignore-max-length
Ignora a verificação de qualidade *Tamanho 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 *Dois pontos não correspondentes*.

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 não coincidentes*.

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 da 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 inalterada*.

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 *Markup XML*.

ignore-xml-invalid
Ignora a verificação de qualidade *Sintaxe XML*.

ignore-zero-width-space
Ignora a verificação de qualidade *Espaçamento nulo*.

ignore-ellipsis
Ignora a verificação de qualidade *Reticências*.

ignore-icu-message-format-syntax
Ignora a verificação de qualidade *Sintaxe de MessageFormat do ICU*.

ignore-long-untranslated
Ignora a verificação de qualidade *Não traduzido há muito tempo*.

ignore-multiple-failures
Ignora a verificação de qualidade *Várias verificações falhadas*.

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 `ignore-*` para qualquer verificação, a usar o identificador dele, para que possa usá-la mesmo para as suas verificações personalizadas.

Essas etiquetas são entendidas tanto nas configurações de *Configuração de componente*, por configurações de cadeias fonte quanto no próprio ficheiro de tradução (por exemplo, no GNU gettext).

2.12.3 Forçar verificações

Novo na versão 3.11.

You can configure a list of checks which can not be ignored by setting *Verificações impostas* in *Configuração de componente*. Each listed check can not be dismissed in the user interface and any string failing this check is marked as *Needs editing* (see *Estados de tradução*).

2.12.4 Gerir letras

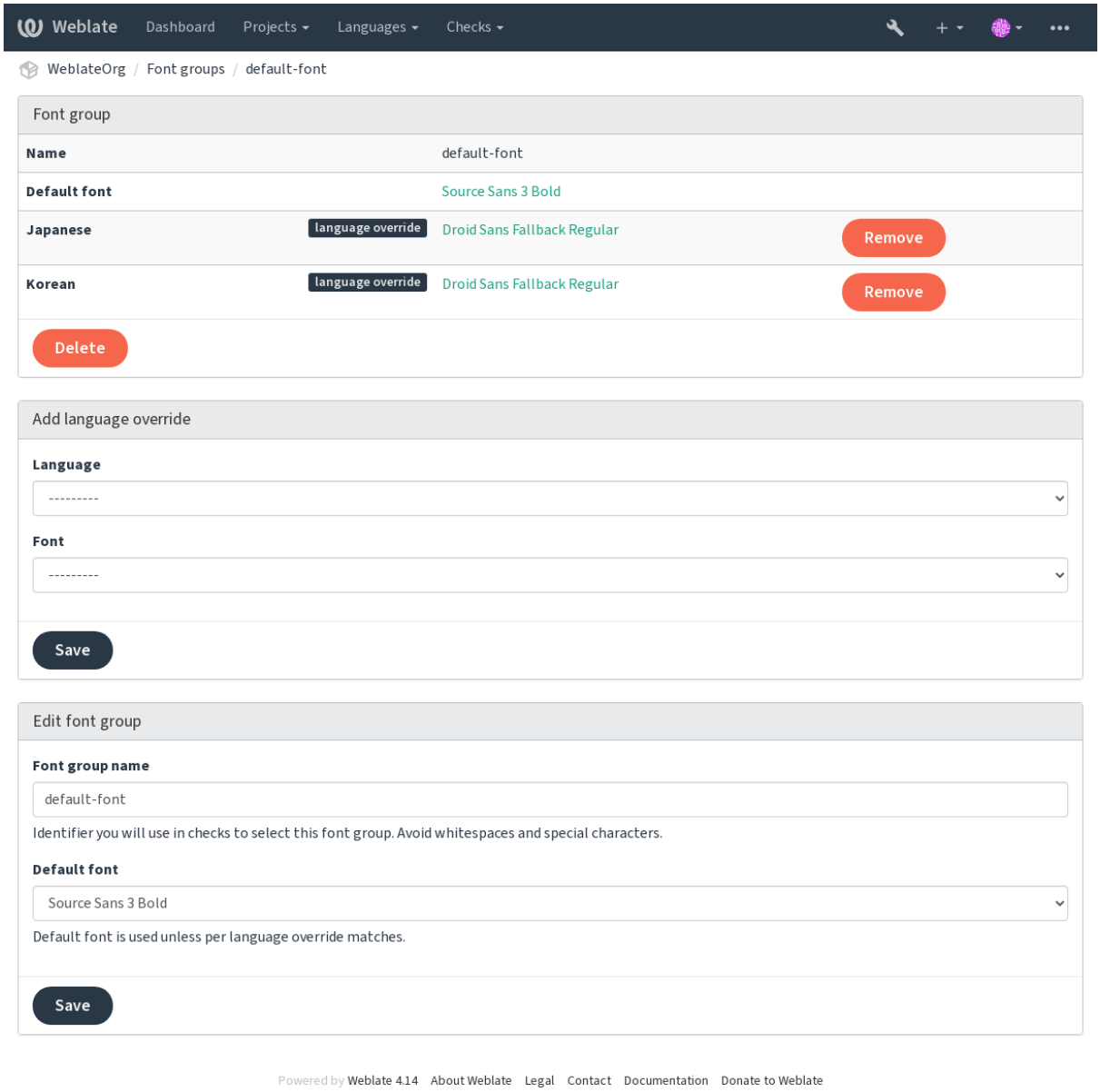
Novo na versão 3.7.

Dica: As fontes carregadas no Weblate são usadas puramente para fins de verificação *Tamanho máximo da tradução*, elas não têm efeito na interface de utilizador do Weblate.

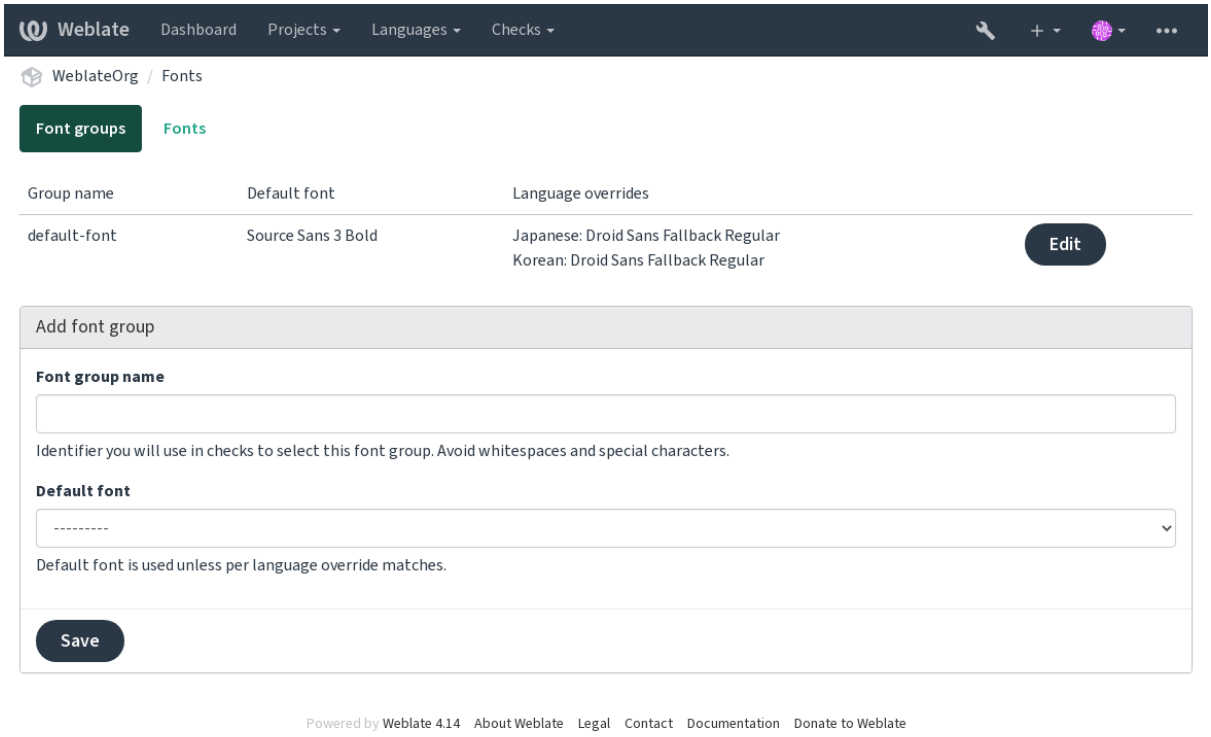
A verificação *Tamanho máximo da tradução* usada para calcular as dimensões do texto renderizado precisa ser carregada no Weblate e selecionada a usar um sinalizador de tradução (ver *Personalizando comportamento a usar marcadores*).

Ferramenta de gestão de fontes do Weblate em *Fonts* sob o menu *Gerir* do seu projeto de tradução fornece interface para carregar e gerir fontes. As fontes TrueType ou OpenType podem ser carregadas, configurar grupos de fontes e usá-los na verificação.

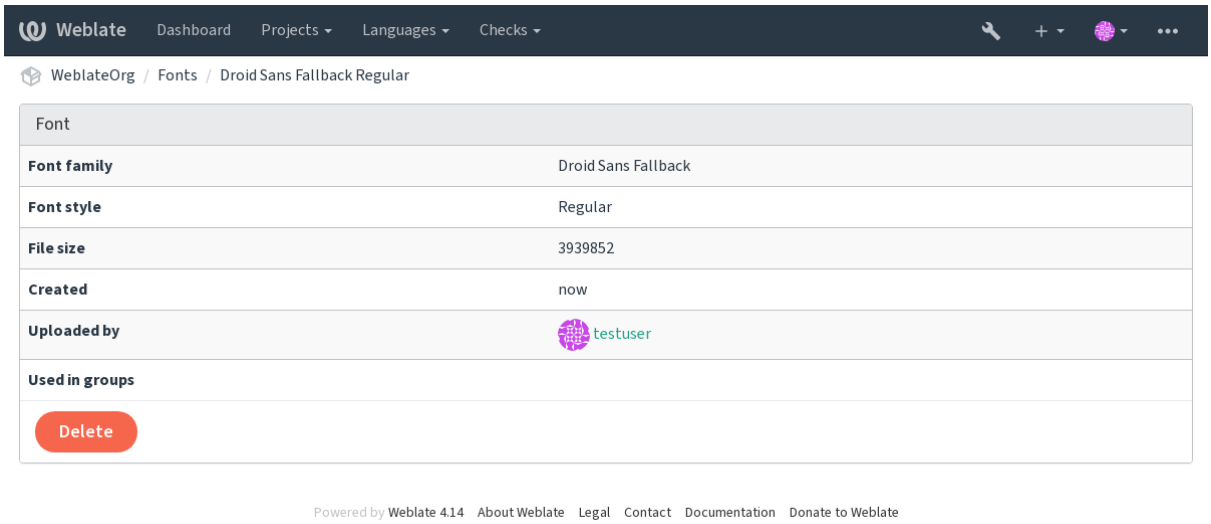
Os grupos de letras permitem definir letras diferentes para idiomas diferentes, o que é normalmente necessário para idiomas não-latinos:



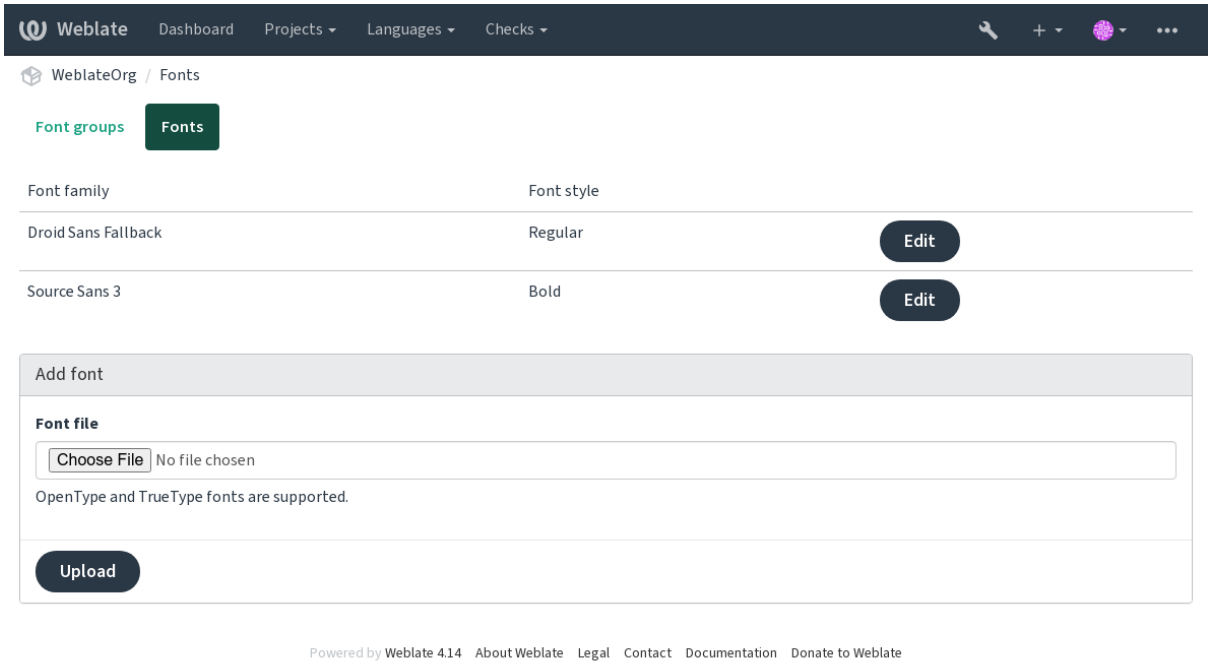
O grupos de letras são identificados pelo nome, que não pode conter espaços ou caracteres especiais, de modo que ele pode ser facilmente utilizado na definição da verificação:



A família de letras e o estilo são automaticamente reconhecidos após carregá-los:



Pode ter muitas letras carregadas para Weblate:



Para usar as letras para verificar o comprimento da cadeia, passe-a os sinalizadores apropriados (veja *Personalizando comportamento a usar marcadores*). Provavelmente precisará dos seguintes:

`max-size:500`
Define o máximo de largura em pixels.

`font-family:ubuntu`
Define o grupo de letras para usar especificando seu identificador.

`font-size:22`
Define o tamanho da fonte em pixels.

2.12.5 Escrever próprias verificações

Uma ampla gama de verificações de qualidade são incorporadas, (veja *Verificações de qualidade*), embora eles possam não cobrir tudo o que deseja verificar. A lista de verificações realizadas pode ser ajustada a usar `CHECK_LIST` e também pode adicionar verificações personalizadas.

1. Crie uma subclasse de *weblate.checks.Check*
2. Define alguns atributos.
3. Implemente o método `check` (se quiser lidar com plurais no seu código) ou o método `check_single` (que faz isso por si).

Alguns exemplos:

Para instalar verificações personalizadas, forneça um caminho totalmente qualificado à classe Python em `CHECK_LIST`, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

Verificar se o texto de tradução não contém «foo»

Esta é uma verificação bastante simples que apenas verifica se a tradução não possui a cadeia «foo».

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
"""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 © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
```

(continues on next page)

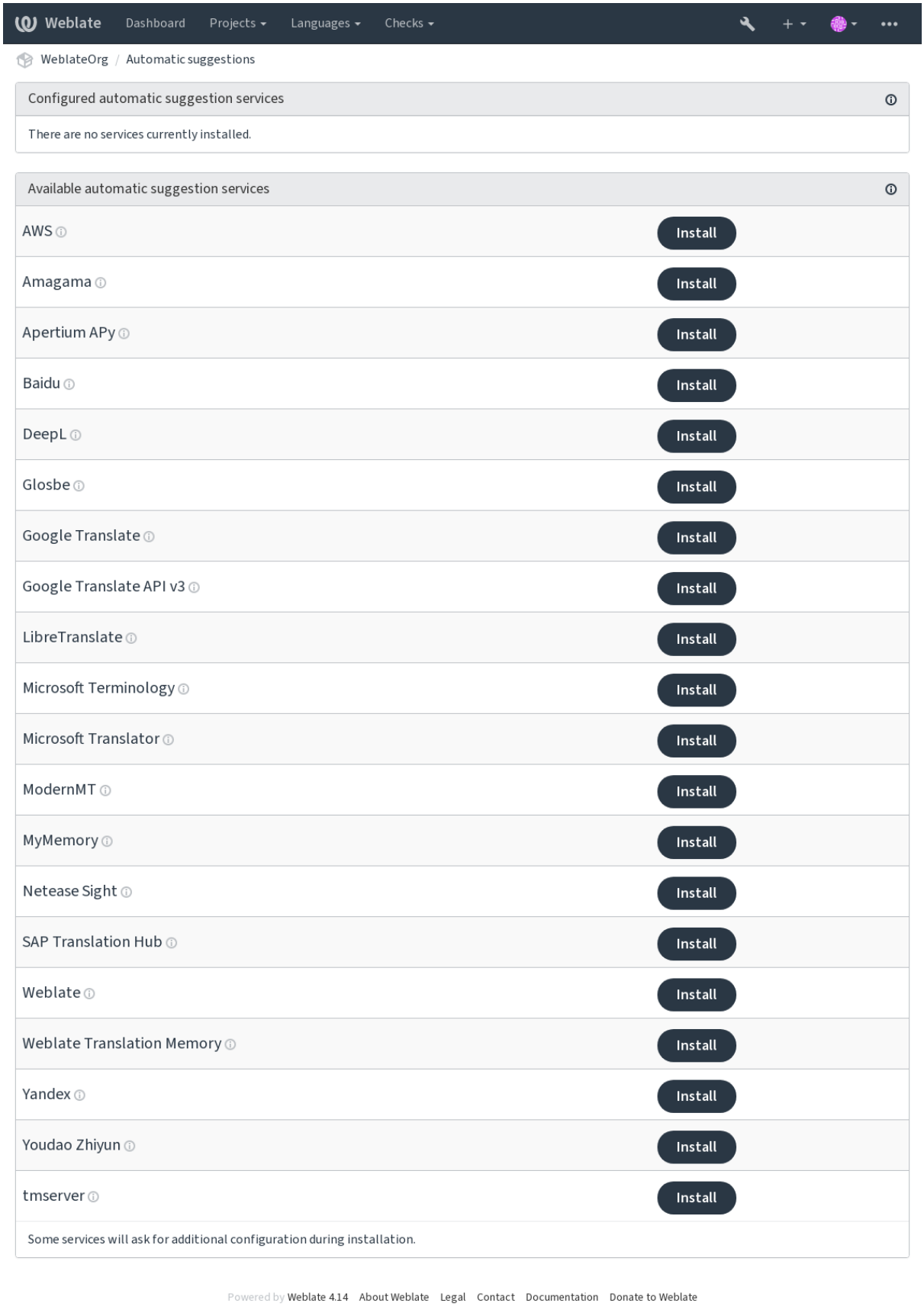
(continuação da página anterior)

```
# 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


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 Configuring automatic suggestions

Alterado na versão 4.13: Antes do Weblate 4.13, os serviços eram configurados no *Configuração*.

O suporte para vários mecanismos de tradução e serviços de memória de tradução (Translation Memory) é integrado. Cada serviço pode ser ativado pelo administrador para todo o site ou nas configurações do projeto:



Nota: Eles estão sujeitos aos seus termos de uso, então certifique-se de ter permissão para usá-los.

The services translate from the source language as configured at *Configuração de componente*, see *Idioma fonte*.

Veja também:

Sugestões automáticas

2.13.1 amaGama

Service ID
`amagama`

Configuração
This service has no configuration.

Special installation of *tmserver* run by the authors of Virtaal.

Veja também:

Installing amaGama, Amagama, Memória de tradução do amaGama

2.13.2 Apertium APy

Service ID
`apertium-apy`

Configuração

`url`	URL da API	

Uma plataforma de tradução de máquina de software livre que fornece traduções para um conjunto limitado de idiomas.

A maneira recomendada de usar o Apertium é executar o seu próprio servidor Apertium-APy.

Veja também:

Apertium website, Apertium APy documentation

2.13.3 AWS

Novo na versão 3.1.

Service ID
`aws`

Configuração

`key`	ID da chave de acesso	
`secret`	Chave secreta da API	
`region`	Nome da região	

O Amazon Translate é um serviço de tradução de máquina neural para traduzir textos de e para o inglês numa variedade de idiomas compatíveis.

Veja também:

Amazon Translate Documentation

2.13.4 Baidu

Novo na versão 3.2.

Service ID
`baidu`

Configuração

`key`	ID de cliente	
`secret`	Chave secreta	

Serviço de tradução de máquina fornecido pelo Baidu.

Este serviço usa uma API e precisa obter um ID e uma chave de API do Baidu para usá-lo.

Veja também:

Baidu Translate API

2.13.5 DeepL

Novo na versão 2.20.

Service ID
`deepl`

Configuração

`url`	URL da API	
`key`	chave API	

DeepL é um serviço pago que fornece boa tradução de máquina para alguns idiomas. Precisa comprar a assinatura do *DeepL API* ou pode usar o plano legado :guilabel:`DeepL Pro (classic) `.

A URL da API para usar com o serviço DeepL. No momento em que este artigo foi escrito, existe a API v1, bem como uma versão gratuita e uma versão paga da API v2.

`https://api.deepl.com/v2/` (padrão no Weblate)
Destina-se ao uso da API no plano pago e a assinatura é baseada em uso.

`https://api-free.deepl.com/v2/`
Destina-se ao uso da API no plano gratuito e a assinatura é baseada em uso.

`https://api.deepl.com/v1/`
Destina-se a ferramentas CAT e é utilizável com assinatura por utilizador.

Anteriormente, o Weblate era classificado como uma ferramenta CAT pelo DeepL, por isso deveria usar a API v1, mas agora é entendido que deve usar a API v2. Portanto, a predefinição é v2 e pode alterar-lo para v1 no caso de ter uma assinatura CAT existente e querer que o Weblate use isso.

A maneira mais fácil de descobrir qual usar é abrir uma URL como a seguinte no seu navegador:

https://api.deepl.com/v2/translate?text=Hello&target_lang=FR&auth_key=XXX

Substitua o XXX pela sua auth_key. Se receber um objeto JSON que contém «Bonjour», tem a URL correta; se não, tente os outros três.

Veja também:

DeepL website, DeepL pricing, DeepL API documentation

2.13.6 Glosbe

Service ID
`glosbe`

Configuração
This service has no configuration.

Dicionário gratuito e memória de tradução para quase todos os idiomas vivos.

The API is gratis to use, but usage of the translations is subject to the license of the used data source. There is a limit of calls that may be done from one IP in a set period of time, to prevent abuse.

Veja também:

Site do Glosbe

2.13.7 Google Tradutor

Service ID
`google-translate`

Configuração

`key`	chave API	

Serviço de tradução de máquina fornecido pelo Google.

Este serviço usa a API do Google Tradutor e precisa obter uma chave de API e ativar a cobrança no console da API do Google.

Veja também:

Google translate documentation

2.13.8 Google Translate API v3

Service ID
`google-translate-api-v3`

Configuração

`credentials`	Informação de conta do serviço Google Translate	
`project`	Projeto Google Translate	
`location`	Localização Google Translate	

Serviço de tradução de máquina fornecido pelos serviços do Google Cloud.

Veja também:

Google translate documentation, Authenticate to Cloud services using client libraries, Creating Google Translate project, Google Cloud App Engine locations

2.13.9 LibreTranslate

Novo na versão 4.7.1.

Service ID
`libretranslate`

Configuração

`url`	URL da API	
`key`	chave API	

LibreTranslate é um serviço gratuito e open-source para traduções automáticas. A instância pública requer uma chave API, mas o LibreTranslate pode ser auto-hospedado e há vários espelhos disponíveis para usar a API gratuitamente.

`https://libretranslate.com/` (instância pública oficial)
Requer uma chave de API para usar fora do website.

Veja também:

LibreTranslate website, LibreTranslate repository, LibreTranslate mirrors

2.13.10 Microsoft Terminology

Novo na versão 2.19.

Service ID
`microsoft-terminology`

Configuração
This service has no configuration.

A API do Serviço de Terminologia Microsoft permite que acesse programaticamente a terminologia, as definições e os textos da interface do utilizador (UI) disponíveis no Portal de Idiomas por meio de um serviço da web.

Veja também:

API do Serviço de Terminologia Microsoft

2.13.11 Microsoft Translator

Novo na versão 2.10.

Service ID
`microsoft-translator`

Configuração

`key`	chave API	
`endpoint_url`	URL do endpoint da aplicação	
`base_url`	URL base da aplicação	Escolhas disponíveis: `api.cognitive.microsofttranslator.com` – Global (non-regional) `api-apc.cognitive.microsofttranslator.com` – Asia Pacific `api-eur.cognitive.microsofttranslator.com` – Europe `api-nam.cognitive.microsofttranslator.com` – North America `api.translator.azure.cn` – China
`region`	Região da aplicação	

Serviço de tradução de máquina fornecido pela Microsoft no portal do Azure como um dos Serviços Cognitivos.

Weblate implementa API V3 do Tradutor.

API V2 de texto do Tradutor

A chave que usa com a API V2 do Tradutor pode ser usada com API 3.

API V3 de texto do Tradutor

You need to register at Azure portal and use the key you obtain there. With new Azure keys, you also need to set `region` to locale of your service.

Dica: Para Azure China, use o desfecho do Portal do Azure.

Veja também:

Cognitive Services - Text Translation API, Microsoft Azure Portal, Base URLs, «Authenticating with a Multi-service resource» «Authenticating with an access token» section

2.13.12 ModernMT

Novo na versão 4.2.

Service ID

`modernmt`

Configuração

`url`	URL da API	
`key`	chave API	

Veja também:

ModernMT API,

2.13.13 MyMemory

Service ID
`mymemory`

Configuração

`email`	E-mail de contacto	
`username`	Nome de utilizador	
`key`	chave API	

Grande memória de tradução com tradução de máquina.

Free, anonymous usage is currently limited to 100 requests/day, or to 1000 requests/day when you provide a contact e-mail address in `email`. You can also ask them for more.

Veja também:

MyMemory website

2.13.14 Netease Sight

Novo na versão 3.3.

Service ID
`netease-sight`

Configuração

`key`	ID de cliente	
`secret`	Chave secreta	

Serviço de tradução de máquina fornecido por NetEase.

Este serviço usa uma API e você precisa obter uma chave e um segredo de NetEase.

Veja também:

NetEase Sight Translation Platform

2.13.15 SAP Translation Hub

Service ID
`sap-translation-hub`

Configuração

`url`	URL da API	
`key`	chave API	
`username`	Nome de utilizador SAP	
`password`	Palavra-passe SAP	
`enable_mt`	Ativar tradução automática	

Serviço de tradução de máquina fornecido por SAP.

Precisa ter uma conta SAP (e o SAP Translation Hub ativado na SAP Cloud Platform) para usar este serviço.

You can also configure whether to also use machine translation services, in addition to the term database.

Nota: To access the Sandbox API, you need to set `url` and `key`.

To access the productive API, you need to set `url`, `username` and `password`.

Veja também:

SAP Translation Hub API, Building the Base URL of SAP Translation Hub

2.13.16 tmserver

Service ID
`tmserver`

Configuração

`url`	URL da API	

Pode executar o seu próprio servidor de memória de tradução usando aquele que vem com o Translate-toolkit e deixar que o Weblate converse com ele. Também pode usá-lo com um servidor amaGama, que é uma versão aprimorada do tmserver.

1. Primeiro, vai importar alguns dados para a memória de tradução:

```
build_tmdb -d /var/lib/tm/db -s en -t cs locale/cs/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t de locale/de/LC_MESSAGES/django.po
build_tmdb -d /var/lib/tm/db -s en -t fr locale/fr/LC_MESSAGES/django.po
```

2. Inicie o tmserver para ouvir as suas solicitações:

```
tmserver -d /var/lib/tm/db
```

3. Configure Weblate to talk to it, the default URL is `http://localhost:8888/tmserver/`.

Veja também:

tmserver Installing amaGama, Amagama, Amagama Translation Memory

2.13.17 Weblate

Service ID
`weblate`

Configuração
This service has no configuration.

O serviço de tradução automática Weblate pode fornecer traduções para textos que já estão traduzidos dentro do Weblate. Ele procura correspondências exatas nos textos existentes.

2.13.18 Weblate Translation Memory

Novo na versão 2.20.

Service ID
`weblate-translation-memory`

Configuração
This service has no configuration.

Use *Memória de Tradução* como um serviço de tradução de máquina. Qualquer texto que tenha sido traduzido no passado (ou carregadalo na memória de tradução) pode ser traduzido dessa maneira.

2.13.19 Yandex

Service ID
`yandex`

Configuração

`key`	chave API	

Serviço de memória de tradução fornecido pela Yandex.

Este serviço usa uma API de tradução e você precisa obter uma chave de API da Yandex.

Veja também:

Yandex Translate API, Powered by Yandex.Translate

2.13.20 Youdao Zhiyun

Novo na versão 3.2.

Service ID
`youdao-zhiyun`

Configuração

`key`	ID de cliente	
`secret`	Chave secreta	

Serviço de tradução de máquina fornecido pelo Youdao.

Este serviço usa uma API e você precisa obter um ID e uma chave de API do Youdao.

Veja também:

Youdao Zhiyun Natural Language Translation Service

2.13.21 Tradução da máquina personalizada

Também pode implementar os seus próprios serviços de tradução automática usando algumas linhas de código Python. Este exemplo implementa a tradução automática numa lista fixa de idiomas usando o módulo Python `dictionary`:

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation


class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"

    def download_languages(self):
        """Return list of languages your machine translation supports."""
        return {"cs"}

    def download_translations(
        self,
        source,
        language,
        text: str,
        unit,
        user,
        search: bool,
        threshold: int = 75,
    ):
        """Return tuple with translations."""
        for t in dictionary.translate(text):
            yield {"text": t, "quality": 100, "service": self.name, "source": text}
```

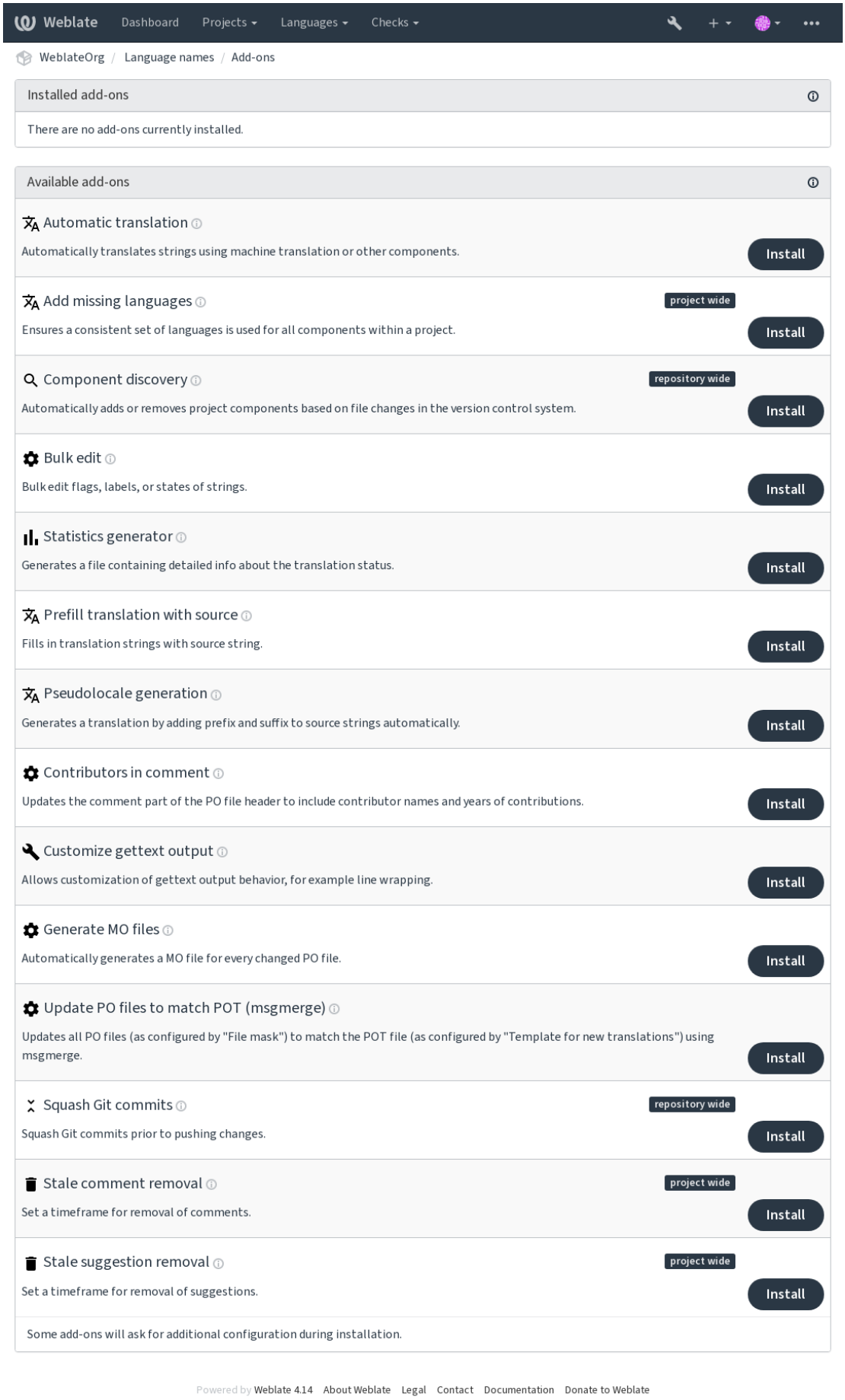
You can list your own class in `WEBLATE_MACHINERY` and Weblate will start using that.

2.14 Extensões

Novo na versão 2.19.

Extensões fornecem maneiras para personalizar e automatizar o fluxo de trabalho de tradução. Administradores podem gerir extensões a partir do menu *Gerir* ↓ *Extensões* dos respectivos componente de tradução.

Dica: Também pode configurar extensões usando *API*, `DEFAULT_ADDONS` ou `install_addon`.


Weblate
Dashboard
Projects
Languages
Checks
WeblateOrg / Language names / Add-ons
Installed add-ons
There are no add-ons currently installed.
Available add-ons
Automatic translation
Automatically translates strings using machine translation or other components.
Install
Add missing languages
project wide
Ensures a consistent set of languages is used for all components within a project.
Install
Component discovery
repository wide
Automatically adds or removes project components based on file changes in the version control system.
Install
Bulk edit
Bulk edit flags, labels, or states of strings.
Install
Statistics generator
Generates a file containing detailed info about the translation status.
Install
Prefill translation with source
Fills in translation strings with source string.
Install
Pseudolocale generation
Generates a translation by adding prefix and suffix to source strings automatically.
Install
Contributors in comment
Updates the comment part of the PO file header to include contributor names and years of contributions.
Install
Customize gettext output
Allows customization of gettext output behavior, for example line wrapping.
Install
Generate MO files
Automatically generates a MO file for every changed PO file.
Install
Update PO files to match POT (msgmerge)
Updates all PO files (as configured by "File mask") to match the POT file (as configured by "Template for new translations") using msgmerge.
Install
Squash Git commits
repository wide
Squash Git commits prior to pushing changes.
Install
Stale comment removal
project wide
Set a timeframe for removal of comments.
Install
Stale suggestion removal
project wide
Set a timeframe for removal of suggestions.
Install
Some add-ons will ask for additional configuration during installation.
Powered by Weblate 4.14 About Weblate Legal Contact Documentation Donate to Weblate

2.14.1 Extensões embutidas

Tradução automática

Novo na versão 3.9.

ID da extensão
`weblate.autotranslate.autotranslate`

Configuração

`mode`	Modo de tradução automático	Escolhas disponíveis: `suggest` – Adiciona como sugestão `translate` – Adiciona como tradução `fuzzy` – Add as «Needing edit»
`filter_type`	Filtro de procura	Por favor, note que a tradução de todas as cadeias irá descartar todas as traduções existentes. Escolhas disponíveis: `all` – Todos os textos `nottranslated` – Cadeias não traduzidas `todo` – Textos não finalizados `fuzzy` – Textos marcados para edição `check: inconsistent` – Verificação com falha: inconsistente
`auto_source`	Source of automated translations	Escolhas disponíveis: `others` – Outros componentes da tradução `mt` – Tradução de máquina
`component`	Componentes	Introduza a lesma de um componente para utilizar como fonte, deixe em branco para utilizar todos os componentes no projeto atual.
`engines`	Motores de tradução automática	
`threshold`	Limite de pontuação	

Gatilhos
Atualização de componente, diária

Traduz automaticamente as cadeias utilizando a tradução automática ou outros componentes.

Ela é acionada:

- Quando novas cadeias aparecem num componente.
- Uma vez por mês para cada componente, isso pode ser configurado a usar `BACKGROUND_TASKS`.

Veja também:

Tradução automática, *Manter traduções iguais entre componentes*

CDN de localização JavaScript

Novo na versão 4.2.

ID da extensão
`weblate.cdn.cdnjs`

Configuração

`threshold`	Limiar de tradução	Limiar para inclusão de traduções.
`css_selector`	Seletor de CSS	Seletor CSS para detectar elementos localizáveis.
`cookie_name`	Nome do cookie do idioma	Nome do cookie que armazena a preferência de idioma.
`files`	Extrair cadeias de ficheiros de HTML	Lista de nomes de ficheiros no repositório atual ou URLs remotas a serem analisados para obter cadeias traduzíveis.

Gatilhos
diário, pós-compromisso do repositório, pós-atualização do repositório

Publica traduções na rede de distribuição de conteúdo para o uso na localização de JavaScript ou HTML.

Pode ser usado para localizar páginas HTML estáticas ou para carregar a localização no código JavaScript.

Gera um URL exclusivo para o seu componente, que pode ser incluído em páginas HTML para localizá-los. Veja weblate-cdn para mais detalhes.

Veja também:

cdn-addon-config, weblate-cdn, cdn-addon-extract, cdn-addon-html

Remover cadeias em branco

Novo na versão 4.4.

ID da extensão
`weblate.cleanup.blank`

Configuração
Esta extensão não tem configuração.

Gatilhos
repositório pós-confirmação, repositório pós-atualização

Remove cadeias não traduzidas dos ficheiros de tradução.

Use isso para não ter cadeias vazias em ficheiros de tradução (por exemplo, se a sua biblioteca de localização os exibe como em falta em vez de recorrer ao texto fonte).

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Limpeza de ficheiros de tradução

ID da extensão
`weblate.cleanup.generic`

Configuração
Esta extensão não tem configuração.

Gatilhos
repositório pré-compromisso, repositório pós-atualização

Atualize todos os ficheiros de tradução para coincidirem com o ficheiro monolingue base. Para a maioria dos formatos de ficheiro, significa remover as chaves de tradução obsoletas que já não existem no ficheiro base.

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Adiciona idiomas em falta

ID da extensão
`weblate.consistency.languages`

Configuração
Esta extensão não tem configuração.

Gatilhos
diário, repositório pós-adição

Garante que um conjunto consistente de idiomas seja usado para todos os componentes de um projeto.

Os idiomas ausentes são verificados uma vez a cada 24 horas e quando novos idiomas são adicionados no Weblate.

Ao contrário da maioria dos outros, esta extensão afeta todo o projeto.

Dica: Traduza as cadeias recém-adicionadas automaticamente com *Tradução automática*.

Descoberta de componentes

ID da extensão
`weblate.discovery.discovery`

Configuração

`match`	Expressão regular para corresponder ficheiros de tradução	
`file_format`	Formato de ficheiro	
`name_template`	Personalizar nome do componente	
`base_file_template`	Definir nome do ficheiro base monolingue	Manter vazio para ficheiros de tradução bilingue.
`new_base_template`	Definir ficheiro base para as novas traduções	Nome do ficheiro utilizado para criar as novas traduções. Para gettext, escolher o ficheiro «.pot».
`intermediate_template`	Ficheiro de idioma intermédio	Nome do ficheiro de tradução intermediária. Geralmente este é um ficheiro de tradução fornecido pelos programadores e é usado na criação de cadeias de origem reais.
`language_regex`	Filtro de idioma	Expressão regular para filtrar ficheiros de tradução ao examinar por uma máscara de ficheiros.
`copy_addons`	Clonar extensões do componente principal para os recém-criados	
`remove`	Remover componentes para os ficheiros inexistentes	
`confirm`	Confirmo que as correspondências acima parecem correctas	

Gatilhos
Pós-atualização do repositório

Adiciona ou remove automaticamente componentes do projeto com base em alterações de ficheiros no sistema de controlo de versão.

Acionada toda vez que o VCS é atualizado, de outra forma semelhante ao comando de gestão `import_project`. Desta forma, pode rastrear vários componentes de tradução dentro de um VCS.

A correspondência é feita a usar expressões regulares que permitem uma configuração complexa, mas é necessário algum conhecimento para fazê-la. Alguns exemplos para casos de uso comum encontram-se na secção de ajuda de extensões.

Uma vez que acertar *Gravar*, uma prévia dos componentes correspondentes será apresentada, de onde pode verificar se a configuração realmente corresponde às suas necessidades:

Weblate Dashboard Projects Languages Checks

WeblateOrg / Language names / Add-ons / Component discovery

Configure add-on

- Please review and confirm the matched components.

Component	Matched files
The following components would be created	
Djangojs	`weblate/locale/hu/LC_MESSAGES/djangojs.po` (hu) `weblate/locale/cs/LC_MESSAGES/djangojs.po` (cs) `weblate/locale/he/LC_MESSAGES/djangojs.po` (he)
Django	`weblate/locale/hu/LC_MESSAGES/django.po` (hu) `weblate/locale/cs/LC_MESSAGES/django.po` (cs) `weblate/locale/he/LC_MESSAGES/django.po` (he)

☐ **I confirm the above matches look correct**

Regular expression to match translation files against

weblate/locale/(?P<language>[^/]*)/LC_MESSAGES/(?P<component>[^/]*)\.po

File format

gettext PO file

Customize the component name

{{ component|title }}

Define the monolingual base filename

Leave empty for bilingual translation files.

Define the base file for new translations

weblate/locale/{{ component }}.pot

Filename of file used for creating new translations. For gettext choose .pot file.

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.

Language filter

^(cs|he|hu)$

Regular expression to filter translation files against when scanning for file mask.

☑ **Clone add-ons from the main component to the newly created ones**

☐ **Remove components for inexistent files**

The regular expression to match translation files has to contain two named groups to match component and language, some examples:

Regular expression	Example matched files	Description

`(?P<language>[^/.]*)/(?P<component>[^/]*)\.po`	`cs/application.po`
	`cs/website.po`
	`de/application.po`
	`de/website.po`
		One folder per language containing translation files for components.

`locale/(?P<language>[^/.]*)/LC_MESSAGES/(?P<component>[^/]*)\.po`	`locale/cs/LC_MESSAGES/application.po`
	`locale/cs/LC_MESSAGES/website.po`
	`locale/de/LC_MESSAGES/application.po`
	`locale/de/LC_MESSAGES/website.po`
		Usual structure for storing gettext PO files.

`src/locale/(?P<component>[^/]*)\.(?P<language>[^/.]*)\.po`	`src/locale/application.cs.po`
	`src/locale/website.cs.po`
	`src/locale/application.de.po`
	`src/locale/website.de.po`
		Using both component and language name within filename.

`locale/(?P<language>[^/.]*)/(?P<component>[^/]*)/(?P=language)\.po`	`locale/cs/application/cs.po`
	`locale/cs/website/cs.po`
	`locale/de/application/de.po`
	`locale/de/website/de.po`
		Using language in both path and filename.

`res/values-(?P<language>[^/.]*)/strings-(?P<component>[^/]*)\.xml`	`res/values-cs/strings-about.xml`
	`res/values-cs/strings-help.xml`
	`res/values-de/strings-about.xml`
	`res/values-de/strings-help.xml`
		Android resource strings, split into several files.

You can use Django template markup in both component name and the monolingual base filename, for example:

`{{ component }}`
Component filename match
`{{ component|title }}`
Component filename with upper case first letter

Save

Powered by Weblate 4.14 About Weblate Legal Contact Documentation Donate to Weblate

Dica: A extensão de descoberta de componentes usa *URLs internas do Weblate*. É uma maneira conveniente de compartilhar a configuração do VCS entre vários componentes. Os componentes vinculados usam o repositório local do componente principal configurado a preencher `weblate://project/main-component` no campo *Repositório do código-fonte* (em *Gerir* ↓ *Configurações* ↓ *Sistema de controle de versão*) de cada componente. Isso economiza tempo com configuração e recursos do sistema também.

Veja também:

Marcação de modelo

Edição em massa

Novo na versão 3.11.

ID da extensão
`weblate.flags.bulk`

Configuração

`q`	Consulta	
`state`	Estado a definir	Escolhas disponíveis: `-1` – Não alterar `10` – Necessita edição `20` – Traduzido `30` – Aprovado
`add_flags`	Bandeiras de tradução para adicionar	
`remove_flags`	Bandeiras de tradução para remover	
`add_labels`	Etiqueta a adicionar	
`remove_labels`	Etiquetas a remover	

Gatilhos
atualização de componente

Edição em série de marcadores, etiquetas, ou estados de cadeias.

Automatize o etiquetação a começar com a consulta de pesquisa `NOT has:label` e adicione etiquetas até que todas as cadeias tenham todas as etiquetas necessárias. Outras operações automatizadas para metadados Weblate também podem ser feitas.

Exemplos::

Table 5: Rotula novas cadeias automaticamente

Consulta de pesquisa	`NOT has:label`
Etiqueta a adicionar	*recente*

Table 6: Marking all Ficheiros de metadados da App Store changelog strings read-only

Consulta de pesquisa	`language:en AND key:changelogs/`
Bandeiras de tradução para adicionar	`read-only`

Veja também:

Edição em massa, *Personalizando comportamento a usar marcadores*, labels

Marcar as traduções inalteradas como «Precisa de edição»

Novo na versão 3.1.

ID da extensão
`weblate.flags.same_edit`

Configuração
Esta extensão não tem configuração.

Gatilhos
unidade pós-criação

Sempre que uma nova cadeia traduzível é importada de VCS e corresponde a uma cadeia fonte, esta é marcada como precisa de edição no Weblate. Isto é especialmente útil para os formatos de ficheiro que incluem cadeias não traduzidas.

Dica: Também pode reforçar a verificação *Tradução inalterada* a adicionar o marcador `strict-same` aos *Marcadores de tradução*.

Veja também:

Estados de tradução

Marcar as novas cadeias fonte como «Precisa de edição»

ID da extensão
`weblate.flags.source_edit`

Configuração
Esta extensão não tem configuração.

Gatilhos
unidade pós-criação

Sempre que uma nova cadeia é importada de VCS, esta é marcada como precisa de edição no Weblate. Deste modo pode filtrar e editar facilmente as cadeias fonte escritas pelos programadores.

Veja também:

Estados de tradução

Marcar as novas traduções como «Precisa de edição»

ID da extensão
`weblate.flags.target_edit`

Configuração
Esta extensão não tem configuração.

Gatilhos
unidade pós-criação

Sempre que uma nova cadeia de tradução é importada de VCS, esta é marcada como precisa de edição no Weblate. Deste modo pode filtrar e editar facilmente as traduções criadas pelos programadores.

Veja também:

Estados de tradução

Gerador de estatísticas

ID da extensão
`weblate.generate.generate`

Configuração

`filename`	Nome do ficheiro gerado	
`template`	Conteúdo do ficheiro gerado	

Gatilhos
repositório pré-compromisso

Gera um ficheiro que contêm a informação detalhada sobre o estado da tradução.

Pode usar um modelo do Django, tanto de nome de ficheiro e conteúdo, veja markdown para uma descrição detalhada de marcação.

Por exemplo, a geração de um ficheiro de resumo para cada tradução:

Nome do ficheiro gerado
`locale/{{ language_code }}.json`

Conteúdo

```
{
   "language": "{{ language_code }}",
   "strings": "{{ stats.all }}",
   "translated": "{{ stats.translated }}",
   "last_changed": "{{ stats.last_changed }}",
   "last_author": "{{ stats.last_author }}",
}
```

Veja também:

Marcação de modelo

Preencher a tradução com fonte

Novo na versão 4.11.

ID da extensão
`weblate.generate.prefill`

Configuração
Esta extensão não tem configuração.

Gatilhos
Atualização de componente, diária

Preenche cadeias de tradução com a cadeia fonte.

Todas as cadeias não traduzidas no componente serão preenchidas com a cadeia de origem e marcadas como precisar a serem editadas. Use isto quando não puder ter cadeias de caracteres vazias nos ficheiros de tradução.

Geração da pseudolocalidade

Novo na versão 4.5.

ID da extensão
`weblate.generate.pseudolocale`

Configuração

`source`	Cadeias fonte	
`target`	Tradução de destino	Todas as cadeias nesta tradução serão sobrescritas
`prefix`	Prefixo de cadeia fixo	
`var_prefix`	Prefixo da cadeia variável	
`suffix`	Sufixo da cadeia fixo	
`var_suffix`	Sufixo da cadeia variável	
`var_multiplier`	Multiplicador de partes variável	Quantas vezes repetir a parte variável dependendo do comprimento do texto fonte.
`include_readonly`	Include read-only strings	

Gatilhos
Atualização de componente, diária

Gera uma tradução a adicionar um prefixo e sufixo às cadeias de origem automaticamente.

Pseudolocalidades são úteis para encontrar textos que não estão preparadas para localização. Isso é feito a alterar todas as cadeias fonte traduzíveis para facilitar a descoberta de textos inalterados ao executar a aplicação no idioma da pseudolocalidade.

Também é possível encontrar cadeias cujas contrapartes localizadas podem não se ajustar ao layout.

O uso das partes variáveis torna possível procurar por textos que podem não caber na interface do utilizador após a localização – isso estende o texto com base no comprimento do texto fonte. As partes variáveis são repetidas pelo comprimento do texto multiplicado pelo multiplicador. Por exemplo `Hello world` com sufixo variável _ e multiplicador variável de 1 se torna `Hello world`_____ – o sufixo é repetido uma vez para cada caractere no texto fonte.

As cadeias serão geradas segundo o padrão:

Prefixo de texto fixo Prefixo de texto variável Texto fonte Sufixo de texto variável Sufixo de texto fixo

---

Dica: Pode usar idiomas reais para teste, mas existem pseudolocalidades disponíveis no Weblate - *en_XA* e *ar_XB*.

Dica: Pode usar esta extensão para iniciar a tradução para uma nova localidade de um idioma existente ou idioma semelhante. Depois de adicionar a tradução ao componente, siga para a extensão. *Exemplo:* Se tem *fr* e deseja iniciar a tradução *fr_CA*, simplesmente defina *fr* como fonte,` fr_CA` como o destino e deixe o prefixo e o sufixo em branco.

Desinstale a extensão depois de preencher a nova tradução para evitar que o Weblate altere as traduções feitas após a cópia.

Contribuintes em comentários

ID da extensão
`weblate.gettext.authors`

Configuração
Esta extensão não tem configuração.

Gatilhos
repositório pré-compromisso

Atualiza a parte do comentário no cabeçalho do ficheiro PO para incluir nomes de colaboradores e anos de contribuições.

O cabeçalho do ficheiro PO terá a seguinte aparência:

```
# Michal Čihař <michal@cihar.com>, 2012, 2018, 2019, 2020.
# Pavel Borecki <pavel@example.com>, 2018, 2019.
# Filip Hron <filip@example.com>, 2018, 2019.
# anonymous <noreply@weblate.org>, 2019.
```

Atualizar a variável ALL_LINGUAS no ficheiro «configure»

ID da extensão
`weblate.gettext.configure`

Configuração
Esta extensão não tem configuração.

Gatilhos
repositório pós-adição, diário

Atualiza a variável ALL_LINGUAS em ficheiros `configure`, `configure.in` ou `configure.ac`, quando uma nova tradução é adicionada.

Personalizar a saída gettext

ID da extensão
`weblate.gettext.customize`

Configuração

`width`	Ajuste de linhas longas	Por predefinição, o gettext quebra as linhas a 77 carateres e novas linhas. Com o parâmetro –no-wrap, quebra apenas em novas linhas. Escolhas disponíveis: `77` – Wrap lines at 77 characters and at newlines (xgettext default) `65535` – Only wrap lines at newlines (like "xgettext –no-wrap") `-1` – Sem quebra de linhas

Gatilhos
pós-carregamento de armazenamento

Permite personalizar o comportamento da saída gettext, por exemplo, a quebra de linhas.

Oferece as seguintes opções:

- Quebrar linhas em 77 caracteres e em novas linhas
- Quebrar as linhas apenas nas novas linhas
- Sem quebra de linhas

Nota: Por padrão, o gettext quebra as linhas em 77 caracteres e em novas linhas. Com o parâmetro `--no-wrap`, a quebra só é feita em novas linhas.

Atualizar ficheiro LINGUAS

ID da extensão
`weblate.gettext.linguas`

Configuração
Esta extensão não tem configuração.

Gatilhos
repositório pós-adição, diário

Atualiza o ficheiro LINGUAS quando é adicionada uma nova tradução.

Gerar ficheiros MO

ID da extensão
`weblate.gettext.mo`

Configuração

`path`	Caminho do ficheiro MO gerado	Se não especificado, será utilizada a localização do ficheiro PO.

Gatilhos
repositório pré-compromisso

Gera automaticamente um ficheiro MO para cada ficheiro PO alterado.

O local do ficheiro MO gerado pode ser personalizada e o campo para ele usa *Marcação de modelo*.

Atualizar ficheiros PO para coincidir com POT (msgmerge)

ID da extensão
`weblate.gettext.msgmerge`

Configuração

`previous`	Manter «msgids» anteriores das cadeias traduzidas	
`no_location`	Remover localizações das cadeias traduzidas	
`fuzzy`	Utilizar correspondência de imprecisa	

Gatilhos
Pós-atualização do repositório

Atualiza todos os ficheiros PO (configurados por *Máscara de ficheiros*) para corresponder ao ficheiro 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.

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Squash de commits git

ID da extensão
`weblate.git.squash`

Configuração

`squash`	Submeter com squash	Escolhas disponíveis: `all` – Todos os commits num só `language` – Por idioma `file` – Por ficheiro `author` – Por autor
`append_trailers`	Anexar linhas finalizadoras à mensagem do commit de squash	As linhas finalizadoras são linhas semelhantes aos cabeçalhos de e-mail RFC 822, no final de uma mensagem de commit de outra forma livre, como «Co-authored-by: …”.
`commit_message`	Mensagem de commit	Esta mensagem de commit será usada em vez das mensagens de commit combinadas dos commits após squash.

Gatilhos
repositório pós-compromisso

Comprimir as submissões Git antes de enviar as alterações.

Commits Git podem ter squash feito antes de fazer o push das alterações num dos seguintes modos:

- Todos os commits num só
- Por idioma
- Por ficheiro
- Por autor

As mensagens de commit originais são mantidas, mas a autoria é perdida a menos que *Por autor* seja selecionada ou a mensagem de commit, seja personalizada para incluí-la.

The original commit messages can optionally be overridden with a custom commit message.

Linhas finalizadoras (linhas de commits como `Co-authored-by:` ...) podem opcionalmente ser removidas das mensagens de commit originais e anexadas ao final da mensagem de compromisso após um squash. Isso também gera crédito próprio `Co-authored-by:` para cada tradutor.

Personalizar a saída JSON

ID da extensão
`weblate.json.customize`

Configuração

`sort_keys`	Ordenar chaves JSON	
`indent`	Indentação JSON	
`style`	Estilo de indentação JSON	Escolhas disponíveis: `spaces` – Espaços `tabs` – Tabulações

Gatilhos
pós-carregamento de armazenamento

Permite ajustar o comportamento da saída JSON, por exemplo, a indentação ou a ordenação.

Format the Java properties file

ID da extensão
`weblate.properties.sort`

Configuração
Esta extensão não tem configuração.

Gatilhos
repositório pré-compromisso

Formats and sorts the Java properties file.

- Consolida novas linhas para Unix.
- Formatação em maiúsculas de sequências de escape Unicode (caso estejam presentes).
- Strips blank lines and comments.
- Sorts the strings by the keys.
- Drops duplicate strings.

Purga de comentários obsoletos

Novo na versão 3.7.

ID da extensão
`weblate.removal.comments`

Configuração

`age`	Dias para manter	

Gatilhos
diário

Definir um prazo para a remoção de comentários.

Isso pode ser útil para remover comentários antigos que podem ter ficado desatualizados. Use com cuidado, pois comentários ficarem velhos não significam que eles perderam importância.

Purga de sugestões obsoletas

Novo na versão 3.7.

ID da extensão
`weblate.removal.suggestions`

Configuração

`age`	Dias para manter	
`votes`	Limites de votação	Limite para a remoção. Este campo não tem efeito com os votos desativados.

Gatilhos
diário

Definir um prazo para a remoção de sugestões.

Pode ser muito útil em relação à votação em sugestão (vej *Revisão por pares*) para remover sugestões que não recebem votos positivos suficientes num determinado período de tempo.

Atualizar ficheiros RESX

Novo na versão 3.9.

ID da extensão
`weblate.resx.update`

Configuração
Esta extensão não tem configuração.

Gatilhos
Pós-atualização do repositório

Atualize todos os ficheiros de tradução para que correspondam ao ficheiro base monolingue original. As cadeias não utilizadas são removidas e as cadeias novas são adicionadas como cópias da cadeia fonte.

Dica: Use *Limpeza de ficheiros de tradução* se só quiser remover chaves de tradução obsoletas.

Veja também:

O Weblate atualiza ficheiros de tradução além de traduções?

Personalizar a saída YAML

Novo na versão 3.10.2.

ID da extensão
`weblate.yaml.customize`

Configuração

`indent`	Indentação de YAML	
`width`	Ajuste de linhas longas	Escolhas disponíveis: `80` – Quebra linhas em 80 caracteres `100` – Quebra linhas em 100 caracteres `120` – Quebra linhas em 120 caracteres `180` – Quebra linhas em 180 caracteres `65535` – Sem quebra de linhas
`line_break`	Quebras de linha	Escolhas disponíveis: `dos` – DOS (\r\n) `unix` – UNIX (\n) `mac` – MAC (\r)

Gatilhos
pós-carregamento de armazenamento

Permite ajustar o comportamento da saída YAML, por exemplo, o comprimento de linha ou novas linhas.

2.14.2 Personalizando a lista de extensões

A lista de extensões é configurada por `WEBLATE_ADDONS`. Para adicionar outra extensão, basta incluir o nome absoluto da classe nesta configuração.

2.14.3 Escrevendo extensões

Também pode escrever as suas próprias extensões. Crie uma subclasse de `weblate.addons.base.BaseAddon` para definir os metadados da extensão e, então, implemente uma função de retorno que vai fazer o processamento.

Veja também:

Desenvolver extensões

2.14.4 Escrevendo scripts para extensões

Extensões também podem ser usadas para executar scripts externos. Isso costumava estar integrado no Weblate, mas agora tem que escrever algum código para embrulhar o seu script com uma extensão.

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
"""Example pre commit script."""


from django.utils.translation import gettext_lazy as _

from weblate.addons.events import EVENT_PRE_COMMIT
from weblate.addons.scripts import BaseScriptAddon


class ExamplePreAddon(BaseScriptAddon):
    # Event used to trigger the script
    events = (EVENT_PRE_COMMIT,)
    # Name of the addon, has to be unique
    name = "weblate.example.pre"
    # Verbose name and long descrption
    verbose = _("Execute script before commit")
    description = _("This add-on executes a script.")

    # Script to execute
    script = "/bin/true"
    # File to add in commit (for pre commit event)
    # does not have to be set
    add_file = "po/{{ language_code }}.po"
```

Para instruções de instalação, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

repositório VCS para qualquer componente.

Além disso, as seguintes variáveis de ambiente estão disponíveis:

WL_VCS
Sistema de controle de versão usado.

WL_REPO
URL do repositório upstream.

WL_PATH
Caminho absoluto ao repositório VCS.

WL_BRANCH
Novo na versão 2.11.

Ramo do repositório configurado no componente atual.

WL_FILEMASK
Máscara de ficheiro para componente atual.

WL_TEMPLATE
Nome de ficheiro de modelo para traduções monolínguas (pode estar vazio).

WL_NEW_BASE
Novo na versão 2.14.

Nome do ficheiro usado para criar novas traduções (pode estar vazio).

WL_FILE_FORMAT
Formato de ficheiro usado no componente atual.

WL_LANGUAGE
Idioma da tradução processada atualmente (não disponível para ganchos de nível de componente).

WL_PREVIOUS_HEAD
HEAD anterior após atualização (disponível apenas ao executar o gancho de pós-atualização).

WL_COMPONENT_SLUG
Novo na versão 3.9.

Slug do componente usado para construir a URL.

WL_PROJECT_SLUG
Novo na versão 3.9.

Slug de projeto usado para construir a URL.

WL_COMPONENT_NAME
Novo na versão 3.9.

Nome de componente.

WL_PROJECT_NAME
Novo na versão 3.9.

Nome do projeto.

WL_COMPONENT_URL
Novo na versão 3.9.

URL do componente.

WL_ENGAGE_URL

Novo na versão 3.9.

URL de engajamento do projeto.

Veja também:

Configuração de componente

Processamento de repositório pós-atualização

Pode ser usado para atualizar ficheiros de tradução quando a fonte VCS do upstream alterar. Para conseguir isso, lembre-se que o Weblate só vê fcheiros dos quais se fez commit com o VCS, então precisa fazer commit das alterações como parte do script.

Por exemplo, com Gulp, pode fazê-lo a usar o código seguinte:

```
#! /bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Processamento pré-commit de traduções

Use o script de commit para alterar automaticamente uma tradução antes de fazer commit dela para o repositório.

É passado como um parâmetro único que consiste o nome de uma tradução atual.

2.15 Memória de Tradução

Novo na versão 2.20.

O Weblate vem com uma memória de tradução integrada que consiste no seguinte:

- Memória de tradução importada manualmente (veja *Interface de utilizador*).
- Traduções armazenadas automaticamente realizadas em Weblate (dependendo dos *Escopos de memória de tradução*).
- Traduções anteriores importadas automaticamente.

O conteúdo da memória de tradução pode ser aplicado de duas maneiras:

- Manualmente, a visão de *Sugestões automáticas* durante a tradução.
- Automaticamente, traduzindo textos usando a extensão *Tradução automática* ou *Tradução automática*.

For installation tips, see *Weblate Translation Memory*, which is turned on by default.

2.15.1 Escopos de memória de tradução

Novo na versão 3.2: Em versões anteriores, a memória de tradução só podia ser carregada de um ficheiro correspondente ao atual escopo da memória de tradução importada.

Os escopos de memória de tradução existem para permitir privacidade e compartilhamento de traduções, de acordo com o comportamento desejado.

Memória de tradução importada

Importar dados de memória de tradução arbitrários usando o comando `import_memory` torna o conteúdo da memória disponível para todos os utilizadores e projetos.

Memória de tradução por utilizador

Armazena todas as traduções do utilizador automaticamente na memória de tradução pessoal de cada respectivo utilizador.

Memória de tradução por projeto

Todas as traduções dentro de um projeto são automaticamente armazenadas em uma memória de tradução de projeto disponível apenas para este projeto.

Memória de tradução compartilhada

Todas as traduções dentro de projetos com memória de tradução compartilhada ativada são armazenadas numa memória de tradução compartilhada disponível para todos os projetos.

Considere cuidadosamente se deve ativar esse recurso para instalações compartilhadas do Weblate, pois pode ter implicações graves:

- As traduções podem ser usadas por qualquer outra pessoa.
- Isso pode levar à divulgação de informações secretas.

2.15.2 A gerir memória de tradução

Interface de utilizador

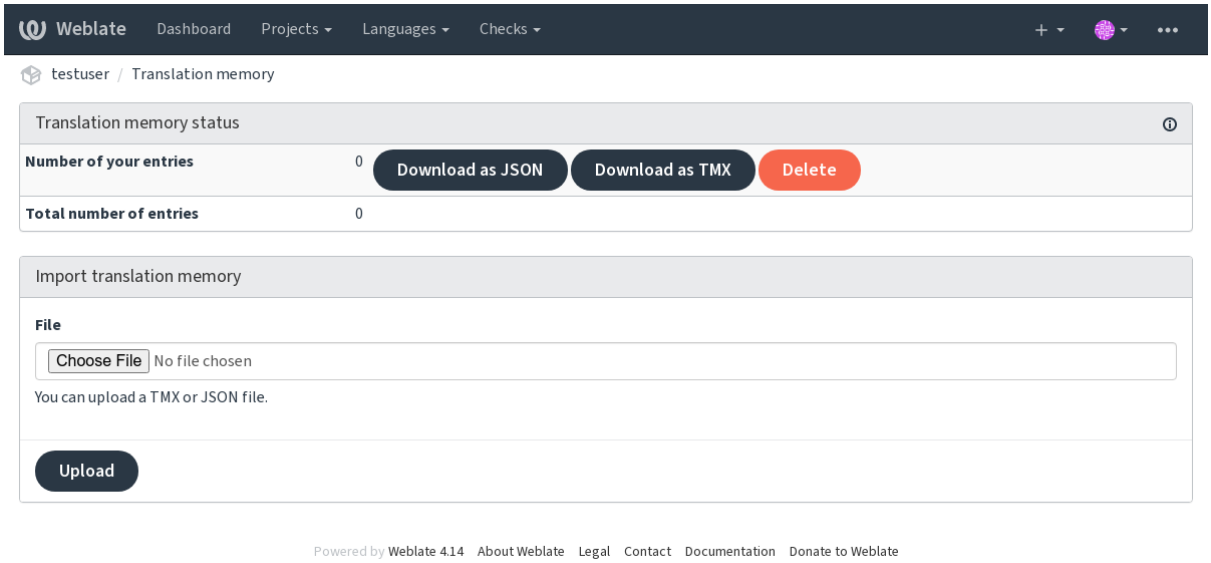
Novo na versão 3.2.

Na interface de utilizador básica, pode gerir memórias de tradução por utilizador e por projeto. Ela pode ser usada para descarregar, limpar ou importar memória de tradução.

Dica: A memória de tradução em JSON pode ser importada ao Weblate, o TMX é fornecido para interoperabilidade com outras ferramentas.

Veja também:

Esquema de memória de tradução Weblate



Interface de gestão

Existem vários comandos de gestão para manipular o conteúdo da memória de tradução. Eles operam na memória de tradução como um todo, não filtrados por escopos (a menos que solicitados por parâmetros):

dump_memory
Exporta a memória para JSON

import_memory
Importa ficheiros TMX ou JSON para a memória de tradução

2.16 Configuração

Todas as configurações estão armazenadas em `settings.py` (como é habitual no Django).

Nota: Após alterar qualquer uma dessas configurações, precisa reiniciar o Weblate - tanto os processos WSGI quanto os Celery.

Caso seja executado como `mod_wsgi:`, precisa reiniciar o Apache para recarregar a configuração.

Veja também:

Verifique também :doc:`Django's documentation <django:ref/settings>`para parâmetros de configuração do próprio Django.

2.16.1 AKISMET_API_KEY

O Weblate pode usar o Akismet para procurar sugestões recebidas anonimamente por spam. Visite akismet.com para comprar uma chave API e associá-la a um site.

2.16.2 ANONYMOUS_USER_NAME

O nome de utilizadores não autenticados.

Veja também:

Controlo de acesso

2.16.3 AUDITLOG_EXPIRY

Novo na versão 3.6.

Quantos dias o Weblate deve manter registos de auditoria, que contêm informações sobre a atividade da conta.

A predefinição é de 180 dias.

2.16.4 AUTH_LOCK_ATTEMPTS

Novo na versão 2.14.

Quantidade máxima de tentativas de autenticação que falharam antes da aplicação da limitação de taxa.

Atualmente, isto é aplicado nos locais seguintes:

- Sign in. Deletes the account password, preventing the user from signing in without requesting a new password.
- Password reset. Prevents new e-mails from being sent, avoiding spamming users with too many password reset attempts.

A predefinição é 10.

Veja também:

Limitação de taxa

2.16.5 AUTO_UPDATE

Novo na versão 3.2.

Alterado na versão 3.11: A opção original de ligar/desligar foi alterada para diferenciar quais cadeias são aceites.

Atualiza todos repositórios diariamente.

Dica: Útil se não estiver a user *Hooks de notificação* para atualizar os repositórios do Weblate automaticamente.

Nota: Existem opções de ligar/desligar, além da seleção de cadeias para compatibilidade com versões anteriores.

As opções são:

`"none"`
Sem atualizações diárias.

`"remote"` e também `False`
Atualizar apenas os repositórios remotos.

`"full"` e também `True`
Atualizar repositórios remotos e mesclar a cópia de trabalho.

Nota: Isto requer que *Tarefas de fundo a usar o Celery* esteja a funcionar e entrará em vigor após ser reiniciado.

2.16.6 AVATAR_URL_PREFIX

Prefixo para construção de URLs de avatars: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Os serviços seguintes funcionam:

Gravatar (predefinição), conforme https://gravatar.com/

```
AVATAR_URL_PREFIX = 'https://www.gravatar.com/'
```

Libravatar, conforme https://www.libravatar.org/

```
AVATAR_URL_PREFIX = 'https://www.libravatar.org/'
```

Veja também:

Cache de avatares, `ENABLE_AVATARS`, *Avatares*

2.16.7 AUTH_TOKEN_VALID

Novo na versão 2.14.

Por quanto tempo o token de autenticação e a palavra-passe temporária dos e-mails de redefinição de palavra-passe são válidos. Definido em número de segundos, usando 172800 (2 dias) como predefinição.

2.16.8 AUTH_PASSWORD_DAYS

Novo na versão 2.15.

Quantos dias a usar a mesma palavra-passe deve ser permitido.

Nota: Mudanças de palavra-passe feitas anteriormente ao Weblate 2.15 não serão consideradas para essa política.

A predefinição é de 180 dias.

2.16.9 AUTOFIX_LIST

Lista de correções automáticas para aplicar ao gravar uma cadeia.

Nota: Forneça um caminho totalmente qualificado à classe Python que implementa a interface de correção automática.

Correções disponíveis:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace`
Corresponde o espaço em branco no início e no fim da cadeia com a fonte.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis`
Substitui pontos ao final (…) se a cadeia fonte tiver um caractere de reticências (…) correspondente.

`weblate.trans.autofixes.chars.RemoveZeroSpace`
Remove caracteres de espaço de largura zero se a fonte não contiver nenhum.

`weblate.trans.autofixes.chars.RemoveControlChars`
Remove caracteres de controle se a fonte não contiver nenhum.

`weblate.trans.autofixes.html.BleachHTML`
Remove a marcação HTML insegura das cadeias sinalizadas como `safe-html` (veja *HTML inseguro*).

Pode selecionar quais usar:

```
AUTOFIX_LIST = (
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
)
```

Veja também:

Correções automáticas, *Correções automáticas personalizadas*

2.16.10 BACKGROUND_TASKS

Novo na versão 4.5.2.

Define com que frequência as tarefas de manutenção demoradas devem ser acionadas para um componente.

No momento, isso controla:

- Extensão *Tradução automática*
- Recálculo de *Verificações e correções*

Escolhas possíveis:

- `monthly` (este é o padrão)
- `weekly`
- `daily`
- `never`

Nota: O aumento da frequência não é recomendado quando o Weblate contém milhares de componentes.

2.16.11 BASIC_LANGUAGES

Novo na versão 4.4.

Lista de idiomas para oferecer aos utilizadores para iniciar uma nova tradução. Quando não especificado, a lista embutida é usada, o que inclui todos os idiomas comumente usados, mas sem variantes específicas de país.

Isso só limita utilizadores não privilegiados a adicionar idiomas indesejados. Os administradores do projeto ainda são apresentados com seleção completa de idiomas definidos no Weblate.

Nota: Isso não define novos idiomas para Weblate, ele filtra apenas os existentes no banco de dados.

Exemplo:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Veja também:

Definições de idioma

2.16.12 BORG_EXTRA_ARGS

Novo na versão 4.9.

Pode passar argumentos adicionais para **borg create** quando backups embutidos forem acionados.

Exemplo:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Veja também:

Fazer backup e mover o Weblate, borg create

2.16.13 CSP_SCRIPT_SRC, CSP_IMG_SRC, CSP_CONNECT_SRC, CSP_STYLE_SRC, CSP_FONT_SRC

Personaliza o cabeçalho `Content-Security-Policy` para Weblate. O cabeçalho é gerado automaticamente com base em integrações ativadas com serviços de terceiros (Matomo, Google Analytics, Sentry, …).

Todos esses tem uma lista vazia como predefinição.

Exemplo:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Veja também:

Política de segurança de conteúdo, Content Security Policy (CSP)

2.16.14 CHECK_LIST

Lista de verificações de qualidade para realizar numa tradução.

Nota: Forneça um caminho totalmente qualificado à classe Python que implementa a interface de verificação.

Ajuste a lista de verificações para incluir as relevantes para si.

Todas as *Verificações de qualidade* embutidas estão ativadas por predefinição, de onde pode alterar essas configurações. Por predefinição, eles são comentados em *Amostra de configuração* para que os valores predefinidos sejam usados. Novas verificações são realizadas para cada versão nova do Weblate.

Pode desativar todas as verificações:

```
CHECK_LIST = ()
```

Pode ativar apenas algumas:

```
CHECK_LIST = (
    "weblate.checks.chars.BeginNewlineCheck",
    "weblate.checks.chars.EndNewlineCheck",
    "weblate.checks.chars.MaxLengthCheck",
)
```

Nota: Alterar esta configuração afeta apenas as traduções recém-alteradas, as verificações existentes ainda serão armazenadas no banco de dados. Para também aplicar alterações nas traduções armazenadas, execute `update-checks`.

Veja também:

Verificações de qualidade, *Personalizando comportamento a usar marcadores*

2.16.15 COMMENT_CLEANUP_DAYS

Novo na versão 3.6.

Apaga comentários após uma determinada quantidade de dias. A predefinição é `None`, ou seja, nada apagado.

2.16.16 COMMIT_PENDING_HOURS

Novo na versão 2.10.

Quantidade de horas entre fazer o commit de alterações pendentes por meio da tarefa de segundo plano.

Veja também:

Configuração de componente, *Idade das alterações a fazer commit*, *Executar tarefas de manutenção*, `commit_pending`

2.16.17 CONTACT_FORM

Novo na versão 4.6.

Configura como o e-mail do formulário de contato está a ser enviado. Escolha uma configuração que corresponda à configuração do seu servidor de e-mail.

"reply-to"
O remetente é usado como `Reply-To`, este é o comportamento padrão.

"from"
O remetente é usado como `From`. O seu servidor de e-mail precisa permitir o envio de tais e-mails.

2.16.18 DATA_DIR

A pasta na qual Weblate armazena todos os dados. Ela contém ligações para repositórios VCS, um índice de texto e vários ficheiros de configuração para ferramentas externas.

Os subdiretórios seguintes geralmente existem:

home
O diretório pessoal usado para invocar scripts.

ssh
Chaves e configuração de SSH.

static
Localização padrão para ficheiros de Django estáticos, especificados por `STATIC_ROOT`. Veja *Servir ficheiros estáticos*.

O contentor do Docker usa um volume separado para isso, veja *Volumes de contentor Docker*.

media
Localização padrão para ficheiros de mídia Django, especificado por `MEDIA_ROOT`. Contém capturas de ecrã enviadas, veja *Contexto visual para cadeias*.

vcs
Repositórios de controle de versão para traduções.

backups
Dados de backup diário. Confira *Dados despejados para backups* para detalhes.

celery
Dados do agendador Celery, consulte *Tarefas de fundo a usar o Celery*.

fonts:
Fontes enviadas pelo utilizador, veja *Gerir letras*.

Nota: Este diretório tem que ser escrito pelo Weblate. Executá-lo como uWSGI significa que o utilizador `www-data` deve ter acesso de escrita.

A maneira mais fácil de conseguir isto é fazer do utilizador o proprietário do diretório:

```
sudo chown www-data:www-data -R $DATA_DIR
```

Defaults to `/home/weblate/data`, but it is expected to be configured.

Veja também:

Permissões do sistema de ficheiros, *Fazer backup e mover o Weblate*

2.16.19 DATABASE_BACKUP

Novo na versão 3.1.

Se os backups de banco de dados devem ser armazenados como texto simples, compactado ou ignorado. Os valores autorizados são:

- `"plain"`
- `"compressed"`
- `"none"`

Veja também:

Fazer backup e mover o Weblate

2.16.20 DEFAULT_ACCESS_CONTROL

Novo na versão 3.3.

A configuração predefinida de controle de acesso para novos projetos:

0
Público

1
Protegido

100
Privado

200
Personalizado

Use *Personalizado* se está a gerir a ACL manualmente, o que significa não confiar na gestão interna do Weblate.

Veja também:

Controlo de acesso ao projeto, *Controlo de acesso*

2.16.21 DEFAULT_AUTO_WATCH

Novo na versão 4.5.

Configura se *Observar automaticamente projetos em contribuição* deve ser ativado para novos utilizadores. O padrão é `True`.

Veja também:

Notificações

2.16.22 DEFAULT_RESTRICTED_COMPONENT

Novo na versão 4.1.

O valor predefinido para a restrição de componentes.

Veja também:

Acesso restrito, *Escopo de grupos*

2.16.23 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DEFAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DEFAULT_MERGE_MESSAGE

Enviar mensagens predefinidas para diferentes operações, consulte *Configuração de componente* para detalhes.

Veja também:

Marcação de modelo, *Configuração de componente*, *Commit, add, delete, merge, add-on, and merge request messages*

2.16.24 DEFAULT_ADDONS

Complementos predefinidos para instalar em cada componente criado.

Nota: Essa configuração afeta apenas componentes recém-criados.

Exemplo:

```
DEFAULT_ADDONS = {
    # Add-on with no parameters
    "weblate.flags.target_edit": {},
    # Add-on with parameters
    "weblate.autotranslate.autotranslate": {
        "mode": "suggest",
        "filter_type": "todo",
        "auto_source": "mt",
        "component": "",
        "engines": ["weblate-translation-memory"],
        "threshold": "80",
    },
}
```

Veja também:

`install_addon`, *Extensões*, `WEBLATE_ADDONS`

2.16.25 DEFAULT_COMMITER_EMAIL

Novo na versão 2.4.

Endereço de e-mail do committer com o padrão `noreply@weblate.org`.

Veja também:

DEFAULT_COMMITER_NAME

2.16.26 DEFAULT_COMMITER_NAME

Novo na versão 2.4.

Nome do committer com o padrão `Weblate`.

Veja também:

DEFAULT_COMMITER_EMAIL

2.16.27 DEFAULT_LANGUAGE

Novo na versão 4.3.2.

Idioma fonte padrão para usar, por exemplo, em *Idioma fonte*.

O padrão é *en*. O objeto de idioma correspondente precisa existir no banco de dados.

Veja também:

Definições de idioma, *Idioma fonte*

2.16.28 DEFAULT_MERGE_STYLE

Novo na versão 3.4.

Mescla o estilo para quaisquer componentes novos.

- *rebase* - predefinição
- *merge*

Veja também:

Configuração de componente, *Estilo de união*

2.16.29 DEFAULT_SHARED_TM

Novo na versão 3.2.

Configura o valor padrão de *Utilizar memória de tradução partilhada* e project-contrib_shared_tm.

2.16.30 DEFAULT_TRANSLATION_PROPAGATION

Novo na versão 2.5.

Configuração predefinida para propagação de tradução, a predefinição é `True`.

Veja também:

Configuração de componente, *Permitir propagação da tradução*

2.16.31 DEFAULT_PULL_MESSAGE

Configura o título e a mensagem padrão para pull requests.

2.16.32 ENABLE_AVATARS

Se se deve ativar avatares baseados em Gravatar para os utilizadores. Por predefinição, isto está ativado.

Avatares são buscados e armazenados em cache no servidor, diminuindo o risco de vazamento de informações privadas, acelerando a experiência do utilizador.

Veja também:

Cache de avatares, `AVATAR_URL_PREFIX`, *Avatares*

2.16.33 ENABLE_HOOKS

Se se deve ativar ganchos remotos anônimos.

Veja também:

Hooks de notificação

2.16.34 ENABLE_HTTPS

Se se deve enviar ligações para Weblate como HTTPS ou HTTP. Esta configuração afeta os e-mails enviados e as URLs absolutas geradas.

Na configuração padrão, este também é usado para várias configurações de Django relacionadas ao HTTPS - ele ativa cookies seguros, ativa/desativa HSTSou ativa redirecionamento para URL HTTPS.

O redirecionamento HTTPS pode ser problemático em alguns casos e pode ter problemas com o redirecionamento infinito no caso de estar a usar um proxy reverso a fazer o encerramento SSL que não passa corretamente os cabeçalhos de protocolo para o Django. Ajuste a sua configuração de proxy reverso para emitir cabeçalhos `X-Forwarded-Proto` ou `Forwarded`, ou configure `SECURE_PROXY_SSL_HEADER` para permitir que o Django detecte corretamente o estado SSL.

Veja também:

`SESSION_COOKIE_SECURE`, `CSRF_COOKIE_SECURE`, `SECURE_SSL_REDIRECT`, `SECURE_PROXY_SSL_HEADER` *Definir domínio correto do site*

2.16.35 ENABLE_SHARING

Ativa/desativa o menu *Compartilhar* para que os utilizadores possam compartilhar o progresso da tradução nas redes sociais.

2.16.36 GET_HELP_URL

Novo na versão 4.5.2.

URL onde o suporte para sua instância do Weblate encontra-se.

2.16.37 GITEA_CREDENTIALS

Novo na versão 4.12.

List for credentials for Gitea servers.

Dica: Use this in case you want Weblate to interact with more of them, for single Gitea endpoint stick with `GITEA_USERNAME` and `GITEA_TOKEN`.

```
GITEA_CREDENTIALS = {
    "try.gitea.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitea.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.38 GITEA_USERNAME

Novo na versão 4.12.

Gitea username used to send pull requests for translation updates.

Veja também:

`GITEA_CREDENTIALS`, *Gitea pull requests*

2.16.39 GITEA_TOKEN

Novo na versão 4.12.

Gitea personal access token used to make API calls to send pull requests for translation updates.

Veja também:

`GITEA_CREDENTIALS`, *Gitea pull requests*, Creating a Gitea personal access token

2.16.40 GITLAB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores de GitLab.

Dica: Use isto no caso de querer que o Weblate interaja com mais deles, para um único ponto final do GitLab com `GITLAB_USERNAME` e `GITLAB_TOKEN`.

```
GITLAB_CREDENTIALS = {
    "gitlab.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitlab.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.41 GITLAB_USERNAME

O nome de utilizador GitLab para enviar merge requests para atualizações de tradução.

Veja também:

`GITLAB_CREDENTIALS`, *Solicitação de mesclagem do GitLab*

2.16.42 GITLAB_TOKEN

Novo na versão 4.3.

GitLab personal access token used to make API calls to send merge requests for translation updates.

Veja também:

`GITLAB_CREDENTIALS`,:ref:*vcs-gitlab*, GitLab: Ficha de acesso pessoal

2.16.43 GITHUB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores GitHub.

Dica: Use isto no caso de querer que o Weblate interaja com mais deles, para um único ponto final do GitHub com `GITHUB_USERNAME` e `GITHUB_TOKEN`.

```
GITHUB_CREDENTIALS = {
    "api.github.com": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "github.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.44 GITHUB_USERNAME

O nome de utilizador GitHub para enviar pull request para atualizações de tradução.

Veja também:

`GITHUB_CREDENTIALS`, *GitHub pull requests*

2.16.45 GITHUB_TOKEN

Novo na versão 4.3.

Token de acesso pessoal GitHub usado para fazer chamadas API para enviar pull requests de tradução.

Veja também:

`GITHUB_CREDENTIALS`, *GitHub pull requests*, Criar um token de acesso pessoal

2.16.46 GOOGLE_ANALYTICS_ID

ID do Google Analytics para ativar o monitoramento do Weblate a usar o Google Analytics.

2.16.47 HIDE_REPO_CREDENTIALS

Oculta credenciais de repositório da interface web. No caso de ter URL do repositório com utilizador e palavra-passe, o Weblate irá ocultá-la quando as informações relacionadas são mostradas aos utilizadores.

Por exemplo, em vez de `https://utilizador:palavra-passe@git.example.com/repo.git`, vai mostrar apenas “”https://git.example.com/repo.git””. Tenta limpar mensagens de erro VCS também de forma semelhante.

Nota: Isso está ativado por predefinição.

2.16.48 HIDE_VERSION

Novo na versão 4.3.1.

Esconde informação de versão de utilizadores não autenticados. Isso também faz todos as ligacões de documentação apontar para a última versão, ao invés da versão de combinação da documentação instalada atualmente.

Ocultar a versão é uma prática de segurança recomendada em algumas empresas, mas não prevê invasores de descobrir a versão a sondar o comportamento.

Nota: Isto está desativado por predefinição.

2.16.49 INTERLEDGER_PAYMENT_POINTERS

Novo na versão 4.12.1.

Lista de Indicadores de Pagamento Interledger (ILPs) para Monetização da Web.

Se vários forem especificados, o compartilhamento de receita probabilístico é obtido selecionando um aleatoriamente.

Consulte <https://webmonetization.org/> para mais detalhes.

Dica: The default value lets users fund Weblate itself.

2.16.50 IP_BEHIND_REVERSE_PROXY

Novo na versão 2.14.

Indica se o Weblate está a ser usado através de um proxy reverso.

Se for definido como `True`, o Weblate obtém o endereço IP de um cabeçalho definido por *IP_PROXY_HEADER*.

Aviso: Certifique-se de que está realmente a usar um proxy reverso e que ele define este cabeçalho, caso contrário, os utilizadores poderão falsificar o endereço IP.

Nota: Isto está ligado por predefinição.

Veja também:

Executar por trás de um proxy reverso, *Limitação de taxa*, *IP_PROXY_HEADER*, *IP_PROXY_OFFSET*

2.16.51 IP_PROXY_HEADER

Novo na versão 2.14.

Indica de qual cabeçalho o Weblate deve obter o endereço IP quando *IP_BEHIND_REVERSE_PROXY* está ativado.

A predefinição é `HTTP_X_FORWARDED_FOR`.

Veja também:

Executar por trás de um proxy reverso, *Limitação de taxa*, `SECURE_PROXY_SSL_HEADER`, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_OFFSET*

2.16.52 IP_PROXY_OFFSET

Novo na versão 2.14.

Indica qual parte de *IP_PROXY_HEADER* é usada como endereço IP do cliente.

Dependendo da configuração, este cabeçalho pode consistir em vários endereços IP (por exemplo, `X-Forwarded-For: a, b, client-ip`) e pode configurar qual endereço do cabeçalho é usado como endereço IP do cliente aqui.

Aviso: Configurar isto afeta a segurança da sua instalação, por isso deve configurá-la só para usar proxies confiáveis para determinar o endereço IP.

A predefinição é 0.

Veja também:

Executar por trás de um proxy reverso, *Limitação de taxa*, SECURE_PROXY_SSL_HEADER, *IP_BEHIND_REVERSE_PROXY*, *IP_PROXY_HEADER*

2.16.53 LEGAL_URL

Novo na versão 3.5.

URL onde a sua instância de Weblate mostra os documentos legais dela.

Dica: Útil se hospeda os seus documentos legais fora do Weblate para incorporá-los ao Weblate, verifique *Legal* para obter detalhes.

Exemplo:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Veja também:

PRIVACY_URL

2.16.54 LICENSE_EXTRA

Licenças adicionais para incluir nas opções de licença.

Nota: Cada definição de licença deve ser uma tupla do seu nome curto, um nome longo e uma URL.

Por exemplo:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

2.16.55 LICENSE_FILTER

Alterado na versão 4.3: A configurar este para valor em branco desativa o alerta de licença.

Filtrar licenças da lista para mostrar. Isto também desativa o alerta de licença quando configurado para vazio.

Nota: Este filtro usa os nomes de licença curtos.

Por exemplo:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

O seguinte desativa o alerta de licença:

```
LICENSE_FILTER = set()
```

Veja também:

alerts

2.16.56 LICENSE_REQUIRED

Define se o atributo de licença em *Configuração de componente* é necessário.

Nota: Isto está desativado por predefinição.

2.16.57 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Whether the length of a given translation should be limited. The restriction is the length of the source string × 10 characters.

Dica: Set this to `False` to allow longer translations (up to 10,000 characters) irrespective of source string length.

Nota: A predefinição é «True».

2.16.58 LOCALIZE_CDN_URL e LOCALIZE_CDN_PATH

Essas configurações definem a extensão *CDN de localização JavaScript*. *LOCALIZE_CDN_URL* define a URL raiz onde o CDN de localização está disponível e *LOCALIZE_CDN_PATH* define o caminho onde o Weblate deve armazenar ficheiros gerados servidos em *LOCALIZE_CDN_URL*.

Dica: O Hosted Weblate usa o `https://weblate-cdn.com/`.

Veja também:

CDN de localização JavaScript

2.16.59 LOGIN_REQUIRED_URLS

A list of URLs you want to require signing in. (Besides the standard rules built into Weblate).

Dica: Isto permite que proteja toda a instalação com uma palavra-passe a usar:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$",)
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Dica: É desejável bloquear o acesso à API também, como mostrado no exemplo acima.

Veja também:

REQUIRE_LOGIN

2.16.60 LOGIN_REQUIRED_URLS_EXCEPTIONS

Lista de exceções para *LOGIN_REQUIRED_URLS*. Se não especificado, os utilizadores podem acessar a página de autenticação.

Algumas das exceções que pode incluir:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$",  # Required for sign in
    r"/static/(.*)$",  # Required for development mode
    r"/widgets/(.*)$",  # Allowing public access to widgets
    r"/data/(.*)$",  # Allowing public access to data exports
    r"/hooks/(.*)$",  # Allowing public access to notification hooks
    r"/api/(.*)$",  # Allowing access to API
    r"/js/i18n/$",  # JavaScript localization
)
```

2.16.61 MATOMO_SITE_ID

ID de um site em Matomo (anteriormente Piwik) que quer rastrear.

Nota: Esta integração não suporta o Matomo Tag Manager.

Veja também:

MATOMO_URL

2.16.62 MATOMO_URL

URL completa (incluindo barra ao final) de uma instalação Matomo (anteriormente Piwik) que deseja usar para rastrear o uso do Weblate. Por favor, consulte <https://matomo.org/> para mais detalhes.

Dica: Esta integração não suporta o Matomo Tag Manager.

Por exemplo:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Veja também:

MATOMO_SITE_ID

2.16.63 NEARBY_MESSAGES

Quantas cadeia devem ser mostradas em torno da cadeia atualmente traduzida. Este é apenas um valor predefinido, os utilizadores podem ajustar-lo em *Perfil do utilizador*.

2.16.64 DEFAULT_PAGE_LIMIT

Novo na versão 4.7.

Número padrão de elementos a serem exibidos quando a paginação está ativa.

2.16.65 PAGURE_CREDENTIALS

Novo na versão 4.3.2.

Lista para credenciais para servidores de Pagure.

Dica: Use isto no caso de querer que o Weblate interaja com mais deles, para um único ponto final do Pagure com `PAGURE_USERNAME` e `PAGURE_TOKEN`.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

2.16.66 PAGURE_USERNAME

Novo na versão 4.3.2.

O nome de utilizador no Pagure para enviar merge requests para atualizações de tradução.

Veja também:

`PAGURE_CREDENTIALS`, *Pagure merge requests*

2.16.67 PAGURE_TOKEN

Novo na versão 4.3.2.

Pagure personal access token used to make API calls to send merge requests for translation updates.

Veja também:

`PAGURE_CREDENTIALS`, *Pagure merge requests*, Pagure API

2.16.68 PRIVACY_URL

Novo na versão 4.8.1.

URL onde a sua instância de Weblate mostra sua política de privacidade.

Dica: Útil se hospeda os seus documentos legais fora do Weblate para incorporá-los ao Weblate, verifique *Legal* para obter detalhes.

Exemplo:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Veja também:

LEGAL_URL

2.16.69 PROJECT_BACKUP_KEEP_COUNT

Novo na versão 4.14.

Defines how many backups per project are kept on the server. It defaults to 3.

Veja também:

Project level backups

2.16.70 PROJECT_BACKUP_KEEP_DAYS

Novo na versão 4.14.

Defines how long the project backups will be kept on the server. Defaults to 30 days.

Veja também:

Project level backups

2.16.71 RATELIMIT_ATTEMPTS

Novo na versão 3.2.

A quantidade máxima de tentativas de autenticação antes da limitação da taxa ser aplicada.

A predefinição é 5.

Veja também:

Limitação de taxa, *RATELIMIT_WINDOW*, *RATELIMIT_LOCKOUT*

2.16.72 RATELIMIT_WINDOW

Novo na versão 3.2.

Por quanto tempo a autenticação é aceita após a limitação da taxa ser aplicada.

Uma quantidade de segundos com a predefinição de 300 (5 minutos).

Veja também:

Limitação de taxa, *RATELIMIT_ATTEMPTS*, *RATELIMIT_LOCKOUT*

2.16.73 RATELIMIT_LOCKOUT

Novo na versão 3.2.

Por quanto tempo a autenticação é bloqueada após a limitação da taxa ser aplicada.

Uma quantidade de segundos com a predefinição de 600 (10 minutos).

Veja também:

Limitação de taxa, *RATELIMIT_ATTEMPTS*, *RATELIMIT_WINDOW*

2.16.74 REGISTRATION_ALLOW_BACKENDS

Novo na versão 4.1.

A lista de backends de autenticação de onde permite o registo. Isso só limita novos registos, os utilizadores ainda se podem autenticar e adicionar autenticação por todos os backends de autenticação configurados.

É recomendado para manter *REGISTRATION_OPEN* ativado enquanto limita os backends de registo, caso contrário, os utilizadores poderão se registar, mas o Weblate não mostrará ligações para se registar na interface do utilizador.

Exemplo:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Dica: Os nomes de backend correspondem aos nomes usados na URL para autenticação.

Veja também:

REGISTRATION_OPEN, *Autenticação*

2.16.75 REGISTRATION_CAPTCHA

Um valor de `True` ou `False` indicando se o registo de contas novas é protegido pelo CAPTCHA. Esta configuração é opcional e uma predifinição de `True` será presumido se não for fornecido.

Se for ativado, um CAPTCHA é adicionado a todas as páginas onde um utilizador digita o endereço de e-mail dele:

- Registo de uma conta nova.
- Recuperação de palavra-passe.
- Adição de uma e-mail a uma conta.
- Formulário de contacto para utilizadores que não estão autenticados.

2.16.76 REGISTRATION_EMAIL_MATCH

Novo na versão 2.17.

Permite filtrar quais endereços de e-mail podem ser registados.

A predefinição é `.*`, que permite que registar qualquer endereço de e-mail.

Pode usá-lo para restringir o registo a um único domínio de e-mail:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.77 REGISTRATION_OPEN

Se o registo de contas novas é atualmente permitido. Esta configuração opcional pode permanecer com a predefinição `True` ou pode ser alterada para `Falsa`.

Esta configuração afeta a autenticação embutida por endereço de e-mail ou através do Python Social Auth (pode listar certos back-ends a usar *REGISTRATION_ALLOW_BACKENDS*).

Nota: Se estiver a usar métodos de autenticação de terceiros, como *Autenticação por LDAP*, ele apenas oculta o formulário de registo, mas novos utilizadores ainda conseguem se autenticar e criar contas.

Veja também:

REGISTRATION_ALLOW_BACKENDS, *REGISTRATION_EMAIL_MATCH*, *Autenticação*

2.16.78 REPOSITORY_ALERT_THRESHOLD

Novo na versão 4.0.2.

Limiar para acionar um alerta para repositórios desatualizados ou aqueles que contêm muitas alterações. A predefinçao é 25.

Veja também:

alerts

2.16.79 REQUIRE_LOGIN

Novo na versão 4.1.

Isso ativa :configuração`URLS_DE_LOGIN_NECESSÁRIOS` e configura o framework REST a requisitar autenticação para todos os pontos finais da API.

Nota: Isto é implementado no *Amostra de configuração*. Para Docker, use *WEBLATE_REQUIRE_LOGIN*.

2.16.80 SENTRY_DSN

Novo na versão 3.9.

DSN do Sentry para usar para *Coletando relatórios de erros*.

Veja também:

Integração Django para o Sentry

2.16.81 IDADE_REGISTRO_SESSÃO_AUTENTICADO

Novo na versão 4.3.

Configura a expiração da sessão para utilizador autenticados. Isso complementa `django:IDADE_REGISTRO_SESSÃO` que é utilizado por utilizadores não autenticados.

Veja também:

`SESSION_COOKIE_AGE`

2.16.82 SIMPLIFY_LANGUAGES

Use códigos de idioma simples para combinações predfinidas de idioma/país. Por exemplo, uma tradução de `fr_FR` usará o código de idioma `fr`. Este é geralmente o comportamento desejado, pois simplifica a lista de idiomas para essas combinações predefinidas.

Desative isto se quiser traduções diferentes para cada variante.

2.16.83 SITE_DOMAIN

Configura o domínio do site. Isso é necessário para produzir ligações absolutas corretas em muitos escopos (por exemplo, ativação de e-mails, notificações ou feeds RSS).

No caso de o Weblate estar a ser executado num porte fora do padrão, inclua-a aqui também.

Exemplos::

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Nota: Esta configuração deve conter apenas o nome de domínio. Para configurar o protocolo (ativar e aplicar HTTPS), use *ENABLE_HTTPS* e para alterar o URL, use *URL_PREFIX*.

Dica: Num contentor Docker, o domínio do site é configurado através de *WEBLATE_ALLOWED_HOSTS*.

Veja também:

Definir domínio correto do site, *Configuração de hosts permitidos*, *Configurar HTTPS corretamente* *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

2.16.84 SITE_TITLE

Título do site a ser usado para o site e e-mails enviados.

2.16.85 SPECIAL_CHARS

Caracteres adicionais para incluir no teclado visual, *Teclado visual*.

O valor predefinido é:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "…")
```

2.16.86 SINGLE_PROJECT

Novo na versão 3.8.

Redireciona os utilizadores diretamente para um projeto ou componente em vez de mostrar o painel. Pode configurá-lo como `True` e, neste caso, só funciona no caso de haver realmente apenas um único projeto no Weblate. Alternativamente, define o projeto e redirecionará incondicionalmente para este projeto.

Alterado na versão 3.11: A configuração agora também aceita um slug de projeto, para forçar a exibição desse único projeto.

Exemplo:

```
SINGLE_PROJECT = "test"
```

2.16.87 SSH_EXTRA_ARGS

Novo na versão 4.9.

Permite adicionar parâmetros personalizados quando o Weblate está a chamar SSH. Isso é útil ao se conectar a servidores a usar criptografia legada ou outros recursos não padrão.

Por exemplo, quando a conexão SSH em Weblate falha com *Unable to negotiate with legacyhost: no matching key exchange method found. Their offer: diffie-hellman-group1-sha1*, pode ativá-lo por entre:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"
```

Dica: O texto é avaliado pelo shell, portanto, certifique-se de pôr aspas em todos os espaços em branco e caracteres especiais.

Veja também:

Opções legadas do OpenSSH

2.16.88 STATUS_URL

A URL onde a sua instância de Weblate relata o estado dela.

2.16.89 SUGGESTION_CLEANUP_DAYS

Novo na versão 3.2.1.

Apaga sugestões automaticamente após uma determinada quantidade de dias. A predefinição é `None`, ou seja, sem exclusões.

2.16.90 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`.

Aviso: A exibição de idiomas pode se tornar inconsistente com isso. As definições de idioma do Weblate se estendem ao longo do tempo e não exibirão o código de idioma para os idiomas definidos.

Veja também:

Definições embutidas de idioma

2.16.91 URL_PREFIX

Esta configuração permite que execute Weblate em algum caminho (caso contrário, depende de ser executado a partir da raiz do servidor web).

Nota: Para usar esta configuração, também precisa configurar o seu servidor para remover este prefixo. Por exemplo, com o WSGI, isso pode ser alcançado definindo `WSGIScriptAlias`.

Dica: O prefixo deve iniciar com um `/`.

Exemplo:

```
URL_PREFIX = "/translations"
```

Nota: Esta configuração não funciona com o servidor embutido do Django, teria que ajustar `urls.py` para conter este prefixo.

2.16.92 VCS_BACKENDS

Configuração de backends VCS disponíveis.

Nota: Weblate tenta usar todos os back-ends suportados para os seus utilizadores.

Dica: Pode limitar escolhas ou adicionar back-ends VCS personalizados a usar isto.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Veja também:

Integração de controlo de versões

2.16.93 VCS_CLONE_DEPTH

Novo na versão 3.10.2.

Configura a profundidade a clonagem de repositórios Weblate deve ter.

Nota: Atualmente, isto só é suportado em *Git*. Por predefinição, o Weblate faz clones rasos dos repositórios para tornar a clonagem mais rápida e economizar espaço no disco. Dependendo do seu uso (por exemplo, ao usar o personalizado *Extensões*), pode aumentar a profundidade ou desligar os clones rasos completamente definindo isso para 0.

Dica: No caso de receber o erro `fatal: protocol error: expected old/new/ref, got 'shallow <hash de commit>'` ao fazer push do Weblate, desative clones rasos completamente configurando:

```
VCS_CLONE_DEPTH = 0
```

2.16.94 WEBLATE_ADDONS

Lista de extensões disponíveis para uso. Para usá-las, elas devem ser ativadas para um determinado componente de tradução. Por padrão, isso inclui todas as extensões embutidas, ao estender a lista, provavelmente pdoe manter as existentes ativadas, por exemplo:

```
WEBLATE_ADDONS = (
    # Built-in add-ons
    "weblate.addons.gettext.GenerateMoAddon",
    "weblate.addons.gettext.UpdateLinguasAddon",
    "weblate.addons.gettext.UpdateConfigureAddon",
    "weblate.addons.gettext.MsgmergeAddon",
    "weblate.addons.gettext.GettextCustomizeAddon",
    "weblate.addons.gettext.GettextAuthorComments",
    "weblate.addons.cleanup.CleanupAddon",
    "weblate.addons.consistency.LangaugeConsistencyAddon",
    "weblate.addons.discovery.DiscoveryAddon",
    "weblate.addons.flags.SourceEditAddon",
    "weblate.addons.flags.TargetEditAddon",
    "weblate.addons.flags.SameEditAddon",
    "weblate.addons.flags.BulkEditAddon",
    "weblate.addons.generate.GenerateFileAddon",
    "weblate.addons.json.JSONCustomizeAddon",
    "weblate.addons.properties.PropertiesSortAddon",
    "weblate.addons.git.GitSquashAddon",
    "weblate.addons.removal.RemoveComments",
    "weblate.addons.removal.RemoveSuggestions",
    "weblate.addons.resx.ResxUpdateAddon",
    "weblate.addons.autotranslate.AutoTranslateAddon",
    "weblate.addons.yaml.YAMLCustomizeAddon",
    "weblate.addons.cdn.CDNJSAddon",
    # Add-on you want to include
    "weblate.addons.example.ExampleAddon",
)
```

Nota: A remoção da extensão da lista não a desinstala dos componentes. Weblate vai travar nesse caso. Por favor, desinstale a extensão de todos os componentes antes de removê-la desta lista.

Veja também:

Extensões, `DEFAULT_ADDONS`

2.16.95 WEBLATE_EXPORTERS

Novo na versão 4.2.

Lista de exportadores disponíveis que oferecem descarregar traduções ou glossários em vários formatos de ficheiro.

Veja também:

Formatos de ficheiros suportados

2.16.96 WEBLATE_FORMATS

Novo na versão 3.0.

Lista de formatos de ficheiro disponíveis para uso.

Nota: A lista predfinida já tem os formatos comuns.

Veja também:

Formatos de ficheiros suportados

2.16.97 WEBLATE_MACHINERY

Novo na versão 4.13.

List of machinery services available for use.

Veja também:

Configuring automatic suggestions

2.16.98 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 *Assinando commits do Git com GnuPG* para mais detalhes.

Veja também:

Assinando commits do Git com GnuPG

2.16.99 WEBSITE_REQUIRED

Define se *Site da Web do Projeto* deve ser especificado ao criar um projeto. Ativado por padrão, pois se adapta às configurações do servidor público.

2.17 Amostra de configuração

O exemplo a seguir é enviado como `weblate/settings_example.py` com Weblate:

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
```

(continues on next page)

(continuação da página anterior)

```
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#


import os
import platform
from logging.handlers import SysLogHandler

# Title of site to use
SITE_TITLE = "Weblate"

# Site domain
SITE_DOMAIN = ""

# Whether site uses https
ENABLE_HTTPS = False

#
# Django settings for Weblate project.
#

DEBUG = True

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

DATABASES = {
    "default": {
        # Use "postgresql" or "mysql".
        "ENGINE": "django.db.backends.postgresql",
        # Database name.
        "NAME": "weblate",
        # Database user.
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password.
        "PASSWORD": "",
        # Set to empty string for localhost.
        "HOST": "127.0.0.1",
        # Set to empty string for default.
        "PORT": "",
        # Customizations for databases.
        "OPTIONS": {
            # In case of using an older MySQL server,
            # which has MyISAM as a default storage
            # "init_command": "SET storage_engine=INNODB",
            # 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:
```

(continues on next page)

(continuação da página anterior)

```
            # "connect_timeout": 28800,
        },
        # Persistent connections
        "CONN_MAX_AGE": 0,
        # Disable server-side cursors, might be needed with pgbouncer
        "DISABLE_SERVER_SIDE_CURSORS": False,
    }
}

# Data directory, you can use following for the development purposes:
# os.path.join(os.path.dirname(os.path.dirname(os.path.abspath(__file__))), "data")
DATA_DIR = "/home/weblate/data"

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "ةيبرعلا"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@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", "□□□"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "□□□"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
```

(continues on next page)

(continuação da página anterior)

```
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "□□□"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "□□□□"),
    ("zh-hant", "□□□□"),
)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# If you set this to False, Django will not 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-generate-secret-key
```

(continues on next page)

(continuação da página anterior)

```
SECRET_KEY = ""

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "OPTIONS": {
            "context_processors": [
                "django.contrib.auth.context_processors.auth",
                "django.template.context_processors.debug",
                "django.template.context_processors.i18n",
                "django.template.context_processors.request",
                "django.template.context_processors.csrf",
                "django.contrib.messages.context_processors.messages",
                "weblate.trans.context_processors.weblate_context",
            ],
        },
        "APP_DIRS": True,
    }
]


# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_USERNAME = None
GITHUB_TOKEN = None

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_USERNAME = None
GITLAB_TOKEN = None

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth2",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_GITHUB_ORG_KEY = ""
SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""

SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = ""
```

(continues on next page)

(continuação da página anterior)

```
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (
    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
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",)
```

(continues on next page)

(continuação da página anterior)

```
SOCIAL_AUTH_SLUGIFY_USERNAMES = True
SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

# Password validation configuration
AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator"  # noqa: E501, pylint: disable=line-too-long
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.CharsPasswordValidator"},
    {"NAME": "weblate.accounts.password_validation.PastPasswordsValidator"},
    # Optional password strength validation by django-zxcvbn-password
    # {
    #     "NAME": "zxcvbn_password.ZXCVBNValidator",
    #     "OPTIONS": {
    #         "min_score": 3,
    #         "user_attributes": ("username", "email", "full_name")
    #     }
    # },
]

# Password hashing (prefer Argon)
PASSWORD_HASHERS = [
    "django.contrib.auth.hashers.Argon2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2PasswordHasher",
    "django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher",
    "django.contrib.auth.hashers.BCryptSHA256PasswordHasher",
]

# Allow new user registrations
REGISTRATION_OPEN = True

# Shortcut for login required setting
REQUIRE_LOGIN = False

# Middleware
MIDDLEWARE = [
    "weblate.middleware.RedirectMiddleware",
    "weblate.middleware.ProxyMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
    "weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
```

(continues on next page)

(continuação da página anterior)

```
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",
    "weblate.screenshots",
    "weblate.fonts",
    "weblate.accounts",
    "weblate.configuration",
    "weblate.utils",
    "weblate.vcs",
    "weblate.wladmin",
    "weblate.metrics",
    "weblate",
    # Optional: Git exporter
    "weblate.gitexport",
    # Standard Django modules
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.admin.apps.SimpleAdminConfig",
    "django.contrib.admindocs",
    "django.contrib.sitemaps",
    "django.contrib.humanize",
    # Third party Django modules
    "social_django",
    "crispy_forms",
    "compressor",
    "rest_framework",
    "rest_framework.authtoken",
    "django_filters",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↪WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - you can also choose "logfile" to log into separate file
#   after configuring it below

# Detect if we can connect to syslog
HAVE_SYSLOG = False
if platform.system() != "Windows":
    try:
        handler = SysLogHandler(address="/dev/log", facility=SysLogHandler.LOG_
↪LOCAL2)
        handler.close()
        HAVE_SYSLOG = True
    except OSError:
```

(continues on next page)

(continuação da página anterior)

```
        HAVE_SYSLOG = False

if DEBUG or not HAVE_SYSLOG:
    DEFAULT_LOG = "console"
else:
    DEFAULT_LOG = "syslog"
DEFAULT_LOGLEVEL = "DEBUG" if DEBUG else "INFO"

# A sample logging configuration. The only tangible logging
# performed by this configuration is to send an email to
# the site admins on every HTTP 500 error when DEBUG=False.
# See http://docs.djangoproject.com/en/stable/topics/logging for
# more details on how to customize your logging configuration.
LOGGING = {
    "version": 1,
    "disable_existing_loggers": True,
    "filters": {"require_debug_false": {"()": "django.utils.log.RequireDebugFalse"}
↪},
    "formatters": {
        "syslog": {"format": "weblate[%(process)d]: %(levelname)s %(message)s"},
        "simple": {"format": "[%(asctime)s: %(levelname)s/%(process)s] %(message)s
↪"},
        "logfile": {"format": "%(asctime)s %(levelname)s %(message)s"},
        "django.server": {
            "()": "django.utils.log.ServerFormatter",
            "format": "[%(server_time)s] %(message)s",
        },
    },
    "handlers": {
        "mail_admins": {
            "level": "ERROR",
            "filters": ["require_debug_false"],
            "class": "django.utils.log.AdminEmailHandler",
            "include_html": True,
        },
        "console": {
            "level": "DEBUG",
            "class": "logging.StreamHandler",
            "formatter": "simple",
        },
        "django.server": {
            "level": "INFO",
            "class": "logging.StreamHandler",
            "formatter": "django.server",
        },
        "syslog": {
            "level": "DEBUG",
            "class": "logging.handlers.SysLogHandler",
            "formatter": "syslog",
            "address": "/dev/log",
            "facility": SysLogHandler.LOG_LOCAL2,
        },
        # Logging to a file
        # "logfile": {
        #     "level":"DEBUG",
        #     "class":"logging.handlers.RotatingFileHandler",
        #     "filename": "/var/log/weblate/weblate.log",
        #     "maxBytes": 100000,
        #     "backupCount": 3,
        #     "formatter": "logfile",
        # },
```

(continues on next page)

(continuação da página anterior)

```
    },
    "loggers": {
        "django.request": {
            "handlers": ["mail_admins", DEFAULT_LOG],
            "level": "ERROR",
            "propagate": True,
        },
        "django.server": {
            "handlers": ["django.server"],
            "level": "INFO",
            "propagate": False,
        },
        # Logging database queries
        # "django.db.backends": {
        #     "handlers": [DEFAULT_LOG],
        #     "level": "DEBUG",
        # },
        "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # Logging VCS operations
        "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # Python Social Auth
        "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # Django Authentication Using LDAP
        "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
        # SAML IdP
        "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    #     "weblate.machinery.apertium.ApertiumAPYTranslation",
    #     "weblate.machinery.baidu.BaiduTranslation",
    #     "weblate.machinery.deepl.DeepLTranslation",
    #     "weblate.machinery.glosbe.GlosbeTranslation",
    #     "weblate.machinery.google.GoogleTranslation",
    #     "weblate.machinery.googlev3.GoogleV3Translation",
    #     "weblate.machinery.libretranslate.LibreTranslateTranslation",
    #     "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    #     "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    #     "weblate.machinery.modernmt.ModernMTTranslation",
    #     "weblate.machinery.mymemory.MyMemoryTranslation",
    #     "weblate.machinery.netease.NeteaseSightTranslation",
    #     "weblate.machinery.tmserver.AmagamaTranslation",
    #     "weblate.machinery.tmserver.TMServerTranslation",
    #     "weblate.machinery.yandex.YandexTranslation",
    #     "weblate.machinery.saptranslationhub.SAPTranslationHub",
    #     "weblate.machinery.youdao.YoudaoTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APy server
MT_APERTIUM_APY = None
```

(continues on next page)

(continuação da página anterior)

```
# DeepL API key
MT_DEEPL_KEY = None

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

# ModernMT
MT_MODERNMT_KEY = None

# MyMemory identification email, see
# https://mymemory.translated.net/doc/spec.php
MT_MYMEMORY_EMAIL = None

# Optional MyMemory credentials to access private translation memory
MT_MYMEMORY_USER = None
MT_MYMEMORY_KEY = None

# Google API key for Google Translate API v2
MT_GOOGLE_KEY = None

# Google Translate API3 credentials and project id
MT_GOOGLE_CREDENTIALS = None
MT_GOOGLE_PROJECT = None

# Baidu app key and secret
MT_BAIDU_ID = None
MT_BAIDU_SECRET = None

# Youdao Zhiyun app key and secret
MT_YOUDAO_ID = None
MT_YOUDAO_SECRET = None

# Netease Sight (Jianwai) app key and secret
MT_NETEASE_KEY = None
MT_NETEASE_SECRET = None

# API key for Yandex Translate API
MT_YANDEX_KEY = None

# tmserver URL
MT_TMSERVER = None

# SAP Translation Hub
MT_SAP_BASE_URL = None
MT_SAP_SANDBOX_APIKEY = None
MT_SAP_USERNAME = None
MT_SAP_PASSWORD = None
MT_SAP_USE_MT = True

# Use HTTPS when creating redirect URLs for social authentication, see
# documentation for more details:
# https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.
↪html#processing-redirects-and-urlopen
SOCIAL_AUTH_REDIRECT_IS_HTTPS = ENABLE_HTTPS
```

(continues on next page)

(continuação da página anterior)

```
# Make CSRF cookie HttpOnly, see documentation for more details:
# https://docs.djangoproject.com/en/1.11/ref/settings/#csrf-cookie-httponly
CSRF_COOKIE_HTTPONLY = True
CSRF_COOKIE_SECURE = ENABLE_HTTPS
# Store CSRF token in session
CSRF_USE_SESSIONS = True
# Customize CSRF failure view
CSRF_FAILURE_VIEW = "weblate.trans.views.error.csrf_failure"
SESSION_COOKIE_SECURE = ENABLE_HTTPS
SESSION_COOKIE_HTTPONLY = True
# SSL redirect
SECURE_SSL_REDIRECT = ENABLE_HTTPS
SECURE_SSL_HOST = SITE_DOMAIN
# Sent referrrer only for same origin links
SECURE_REFERRER_POLICY = "same-origin"
# SSL redirect URL exemption list
SECURE_REDIRECT_EXEMPT = (r"healthz/$",)  # Allowing HTTP access to health check
# Session cookie age (in seconds)
SESSION_COOKIE_AGE = 1000
SESSION_COOKIE_AGE_AUTHENTICATED = 1209600
SESSION_COOKIE_SAMESITE = "Lax"
# Increase allowed upload size
DATA_UPLOAD_MAX_MEMORY_SIZE = 50000000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
```

(continues on next page)

(continuação da página anterior)

```
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# By default the length of a given translation is limited to the length of
# the source string * 10 characters. Set this option to False to allow longer
# translations (up to 10.000 characters)
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH = True

# Use simple language codes for default language/country combinations
SIMPLIFY_LANGUAGES = True

# Render forms using bootstrap
CRISPY_TEMPLATE_PACK = "bootstrap3"

# List of quality checks
# CHECK_LIST = (
#     "weblate.checks.same.SameCheck",
#     "weblate.checks.chars.BeginNewlineCheck",
#     "weblate.checks.chars.EndNewlineCheck",
#     "weblate.checks.chars.BeginSpaceCheck",
#     "weblate.checks.chars.EndSpaceCheck",
#     "weblate.checks.chars.DoubleSpaceCheck",
#     "weblate.checks.chars.EndStopCheck",
#     "weblate.checks.chars.EndColonCheck",
#     "weblate.checks.chars.EndQuestionCheck",
#     "weblate.checks.chars.EndExclamationCheck",
#     "weblate.checks.chars.EndEllipsisCheck",
#     "weblate.checks.chars.EndSemicolonCheck",
#     "weblate.checks.chars.MaxLengthCheck",
#     "weblate.checks.chars.KashidaCheck",
#     "weblate.checks.chars.PunctuationSpacingCheck",
#     "weblate.checks.format.PythonFormatCheck",
#     "weblate.checks.format.PythonBraceFormatCheck",
#     "weblate.checks.format.PHPFormatCheck",
#     "weblate.checks.format.CFormatCheck",
#     "weblate.checks.format.PerlFormatCheck",
#     "weblate.checks.format.JavaScriptFormatCheck",
#     "weblate.checks.format.LuaFormatCheck",
#     "weblate.checks.format.ObjectPascalFormatCheck",
#     "weblate.checks.format.SchemeFormatCheck",
#     "weblate.checks.format.CSharpFormatCheck",
#     "weblate.checks.format.JavaFormatCheck",
#     "weblate.checks.format.JavaMessageFormatCheck",
#     "weblate.checks.format.PercentPlaceholdersCheck",
#     "weblate.checks.format.VueFormattingCheck",
#     "weblate.checks.format.I18NextInterpolationCheck",
#     "weblate.checks.format.ESTemplateLiteralsCheck",
#     "weblate.checks.angularjs.AngularJSInterpolationCheck",
#     "weblate.checks.icu.ICUMessageFormatCheck",
#     "weblate.checks.icu.ICUSourceCheck",
#     "weblate.checks.qt.QtFormatCheck",
#     "weblate.checks.qt.QtPluralCheck",
#     "weblate.checks.ruby.RubyFormatCheck",
#     "weblate.checks.consistency.PluralsCheck",
#     "weblate.checks.consistency.SamePluralsCheck",
#     "weblate.checks.consistency.ConsistencyCheck",
```

(continues on next page)

(continuação da página anterior)

```
#     "weblate.checks.consistency.TranslatedCheck",
#     "weblate.checks.chars.EscapedNewlineCountingCheck",
#     "weblate.checks.chars.NewLineCountCheck",
#     "weblate.checks.markup.BBCodeCheck",
#     "weblate.checks.chars.ZeroWidthSpaceCheck",
#     "weblate.checks.render.MaxSizeCheck",
#     "weblate.checks.markup.XMLValidityCheck",
#     "weblate.checks.markup.XMLTagsCheck",
#     "weblate.checks.markup.MarkdownRefLinkCheck",
#     "weblate.checks.markup.MarkdownLinkCheck",
#     "weblate.checks.markup.MarkdownSyntaxCheck",
#     "weblate.checks.markup.URLCheck",
#     "weblate.checks.markup.SafeHTMLCheck",
#     "weblate.checks.placeholders.PlaceholderCheck",
#     "weblate.checks.placeholders.RegexCheck",
#     "weblate.checks.duplicate.DuplicateCheck",
#     "weblate.checks.source.OptionalPluralCheck",
#     "weblate.checks.source.EllipsisCheck",
#     "weblate.checks.source.MultipleFailingCheck",
#     "weblate.checks.source.LongUntranslatedCheck",
#     "weblate.checks.format.MultipleUnnamedFormatsCheck",
#     "weblate.checks.glossary.GlossaryCheck",
# )

# List of automatic fixups
# AUTOFIX_LIST = (
#     "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",
#     "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",
#     "weblate.trans.autofixes.chars.RemoveZeroSpace",
#     "weblate.trans.autofixes.chars.RemoveControlChars",
# )

# List of enabled addons
# WEBLATE_ADDONS = (
#     "weblate.addons.gettext.GenerateMoAddon",
#     "weblate.addons.gettext.UpdateLinguasAddon",
#     "weblate.addons.gettext.UpdateConfigureAddon",
#     "weblate.addons.gettext.MsgmergeAddon",
#     "weblate.addons.gettext.GettextCustomizeAddon",
#     "weblate.addons.gettext.GettextAuthorComments",
#     "weblate.addons.cleanup.CleanupAddon",
#     "weblate.addons.cleanup.RemoveBlankAddon",
#     "weblate.addons.consistency.LangaugeConsistencyAddon",
#     "weblate.addons.discovery.DiscoveryAddon",
#     "weblate.addons.autotranslate.AutoTranslateAddon",
#     "weblate.addons.flags.SourceEditAddon",
#     "weblate.addons.flags.TargetEditAddon",
#     "weblate.addons.flags.SameEditAddon",
#     "weblate.addons.flags.BulkEditAddon",
#     "weblate.addons.generate.GenerateFileAddon",
#     "weblate.addons.generate.PseudolocaleAddon",
#     "weblate.addons.generate.PrefillAddon",
#     "weblate.addons.json.JSONCustomizeAddon",
#     "weblate.addons.properties.PropertiesSortAddon",
#     "weblate.addons.git.GitSquashAddon",
#     "weblate.addons.removal.RemoveComments",
#     "weblate.addons.removal.RemoveSuggestions",
#     "weblate.addons.resx.ResxUpdateAddon",
#     "weblate.addons.yaml.YAMLCustomizeAddon",
#     "weblate.addons.cdn.CDNJSAddon",
# )
```

(continues on next page)

(continuação da página anterior)

```
# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

# List of URLs your site is supposed to serve
ALLOWED_HOSTS = ["*"]

# Configuration for caching
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/1",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # "LOCATION": "unix:///var/run/redis/redis.sock?db=1",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
            # If you set password here, adjust CELERY_BROKER_URL as well
            "PASSWORD": None,
            "CONNECTION_POOL_KWARGS": {},
        },
        "KEY_PREFIX": "weblate",
    },
    "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"},
```

(continues on next page)

(continuação da página anterior)

```
    "DEFAULT_PAGINATION_CLASS": ("rest_framework.pagination.PageNumberPagination"),
    "PAGE_SIZE": 20,
    "VIEW_DESCRIPTION_FUNCTION": "weblate.api.views.get_view_description",
    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL
FONTS_CDN_URL = None

# Django compressor offline mode
COMPRESS_OFFLINE = False
COMPRESS_OFFLINE_CONTEXT = [
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI":␣
↪True},
    {"fonts_cdn_url": FONTS_CDN_URL, "STATIC_URL": STATIC_URL, "LANGUAGE_BIDI":␣
↪False},
]

# Require login for all URLs
if REQUIRE_LOGIN:
    LOGIN_REQUIRED_URLS = (r"/(.*)$",)

# In such case you will want to include some of the exceptions
# LOGIN_REQUIRED_URLS_EXCEPTIONS = (
#     rf"{URL_PREFIX}/accounts/(.*)$",  # Required for login
#     rf"{URL_PREFIX}/admin/login/(.*)$",  # Required for admin login
#     rf"{URL_PREFIX}/static/(.*)$",  # Required for development mode
#     rf"{URL_PREFIX}/widgets/(.*)$",  # Allowing public access to widgets
#     rf"{URL_PREFIX}/data/(.*)$",  # Allowing public access to data exports
#     rf"{URL_PREFIX}/hooks/(.*)$",  # Allowing public access to notification hooks
#     rf"{URL_PREFIX}/healthz/$",  # Allowing public access to health check
#     rf"{URL_PREFIX}/api/(.*)$",  # Allowing access to API
#     rf"{URL_PREFIX}/js/i18n/$",  # JavaScript localization
#     rf"{URL_PREFIX}/contact/$",  # Optional for contact form
#     rf"{URL_PREFIX}/legal/(.*)$",  # Optional for legal app
#     rf"{URL_PREFIX}/avatar/(.*)$",  # Optional for avatars
# )

# Silence some of the Django system checks
SILENCED_SYSTEM_CHECKS = [
    # We have modified django.contrib.auth.middleware.AuthenticationMiddleware
    # as weblate.accounts.middleware.AuthenticationMiddleware
    "admin.E408"
]

# Celery worker configuration for testing
# CELERY_TASK_ALWAYS_EAGER = True
# CELERY_BROKER_URL = "memory://"
# CELERY_TASK_EAGER_PROPAGATES = True
# Celery worker configuration for production
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL

# Celery settings, it is not recommended to change these
CELERY_WORKER_MAX_MEMORY_PER_CHILD = 200000
CELERY_BEAT_SCHEDULE_FILENAME = os.path.join(DATA_DIR, "celery", "beat-schedule")
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate*": {"queue": "translate"},
    "weblate.accounts.tasks.notify_*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
```

(continues on next page)

(continuação da página anterior)

```
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

# Enable plain database backups
DATABASE_BACKUP = "plain"

# Enable auto updating
AUTO_UPDATE = False

# PGP commits signing
WEBLATE_GPG_IDENTITY = None

# Third party services integration
MATOMO_SITE_ID = None
MATOMO_URL = None
GOOGLE_ANALYTICS_ID = None
SENTRY_DSN = None
SENTRY_ENVIRONMENT = SITE_DOMAIN
AKISMET_API_KEY = None
```

2.18 Comandos de gerência

Nota: Executar comandos de gestão sob um utilizador diferente daquele que executa o seu servidor web pode resultar em ficheiros a receber permissões erradas, por favor, confira *Permissões do sistema de ficheiros* para mais detalhes.

Encontrará comandos básicos de gestão (disponíveis como `./manage.py` nas fontes do Django ou como um conjunto estendido num script chamado **`weblate`** instalável em cima do Weblate).

2.18.1 Invocando comandos de gestão

Como mencionado anteriormente, a invocação depende de como instalou o Weblate.

Se estiver a usar o virtualenv para Weblate, pode especificar o caminho completo para **`weblate`** ou ativar o virtualenv antes de invocá-lo:

```
# Direct invocation
~/weblate-env/bin/weblate

# Activating virtualenv adds it to search path
. ~/weblate-env/bin/activate
weblate
```

Se estiver a usar o código-fonte diretamente (de um tarball ou checkout Git), o script de gestão é `. /manage.py` disponível nos fontes do Weblate. Para executá-lo:

```
python ./manage.py list_versions
```

If you've installed Weblate using the pip installer, or by using the `./setup.py` script, the **`weblate`** is installed to your path (or virtualenv path), from where you can use it to control Weblate:

```
weblate list_versions
```

Para a imagem Docker, o script é instalado como acima e pode executá-lo usando **docker exec**:

```
docker exec --user weblate <container> weblate list_versions
```

Para **docker-compose** o processo é semelhante, apenas tem que usar **docker-compose exec**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

Caso precise passar-le um ficheiro, pode adicionar temporariamente um volume:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/
↪users.json
```

Veja também:

Instalando a usar Docker, *Instalando no Debian e Ubuntu*, *Instalando no SUSE e openSUSE*, *Instalando no RedHat, Fedora e CentOS*, *Instalando a partir do código-fonte*

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Novo na versão 2.5.

Importa uma tradução do fcheiro para usar como sugestão para a tradução fornecida. Pula traduções duplicadas; apenas os diferentes são adicionados.

--author USER@EXAMPLE.COM

E-mail do autor para sugestões. Este utilizador deve existir antes da importação (você pode criar um na interface administrativa, se necessário).

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.

Executa tradução automática com base em outras traduções de componentes.

--source PROJECT/COMPONENT

Especifica o componente a ser usado como fonte disponível para tradução. Se não for especificado, todos os componentes do projeto serão usados.

--user USERNAME

Especifica o nome de utilizador listado como autor das traduções. «Utilizador anônimo» é usado se não for especificado.

--overwrite

Se deve sobrescrever as traduções existentes.

--inconsistent

Se deve substituir as traduções existentes que são inconsistentes (veja *Inconsistente*).

--add

Adiciona automaticamente o idioma se uma determinada tradução não existir.

--mt MT

Use a tradução automática em vez de outros componentes como traduções de máquina.

--threshold THRESHOLD

Limite de similaridade para tradução de máquina, a predefinição é 80.

--mode MODE

Especifica o modo de tradução tendo como o padrão `translate`, mas `fuzzy` ou `suggest` pode ser usado.

Exemplo:

```
weblate auto_translate --user nijel --inconsistent --source weblate/application␣
↪weblate website cs
```

Veja também:

Tradução automática

2.18.4 celery_queues

weblate celery_queues

Novo na versão 3.7.

Exibe o tamanho das filas de tarefas do Celery.

2.18.5 checkgit

weblate checkgit <project|project/component>

Imprime o estado atual do repositório Git de backend.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

2.18.6 commitgit

weblate commitgit <project|project/component>

Faz commit de todas as alterações pendentes possíveis para o repositório Git de backend.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Faz commit de alterações pendentes anteriores a uma determinada idade.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

--age HOURS

Idade em horas para fazer commit. Se não for especificado, o valor configurado em *Configuração de componente* é usado.

Nota: Isso é realizado automaticamente em segundo plano pelo Weblate, de modo que não há necessidade real de invocar isso manualmente, além de forçar um commit anterior do que o especificado por *Configuração de componente*.

Veja também:

Executar tarefas de manutenção, `COMMIT_PENDING_HOURS`

2.18.8 cleanuptrans

weblate cleanuptrans

Limpa verificações e sugestões de tradução órfãs. Normalmente, não há necessidade de executá-lo manualmente, pois as limpezas acontecem automaticamente em segundo plano.

Veja também:

Executar tarefas de manutenção

2.18.9 cleanup_ssh_keys

weblate cleanup_ssh_keys

Novo na versão 4.9.1.

Executa a remoção das chaves de host SSH armazenadas:

- Remove as chaves RSA descontinuadas do GitHub que podem causar problemas de conexão com o GitHub.
- Remove entradas duplicadas nas chaves do host.

Veja também:

Repositórios SSH

2.18.10 createadmin

weblate createadmin

Cria uma conta `admin` com uma palavra-passe aleatória, a menos que seja especificada.

--password PASSWORD

Fornece uma palavra-passe na linha de comando, para não gerar uma palavra-passe aleatória.

--no-password

Não define a palavra-passe, isso pode ser útil com *–update*.

--username USERNAME

Use o nome dado em vez de `admin`.

--email USER@EXAMPLE.COM

Especifica o endereço de e-mail do administrador.

--name

Especifica o nome (visível) do administrador.

--update

Atualiza o utilizador existente (pode usar isto para alterar palavras-passe).

Alterado na versão 2.9: Adicionados os parâmetros `--username`, `--email`, `--name` e `--update`.

2.18.11 dump_memory

weblate dump_memory

Novo na versão 2.20.

Exporta um ficheiro JSON que contem o conteúdo de memória de tradução do Weblate.

Veja também:

Memória de Tradução, *Esquema de memória de tradução Weblate*

2.18.12 dumpuserdata

weblate dumpuserdata <file.json>

Dumps userdata to a file for later use by `importuserdata`.

Dica: Isso é útil ao migrar ou mesclar instâncias do Weblate.

2.18.13 import_demo

weblate import_demo

Novo na versão 4.1.

Creates a demo project with components based on <https://github.com/WeblateOrg/demo>. Make sure the celery tasks are running before running this command.

Isso pode ser útil ao desenvolver Weblate.

2.18.14 import_json

weblate import_json <json-file>

Novo na versão 2.7.

Importação em lote de componentes com base em dados JSON.

A estrutura do ficheiro JSON importado corresponde basicamente ao objeto do componente (consulte *GET /api/components/(string:project)/(string:component)/*). Deve incluir os campos `name` e `filemask`.

--project `PROJECT`

Especifica de onde os componentes serão importados.

--main-component `COMPONENT`

Usa o repositório VCS fornecido deste componente para todos eles.

--ignore

Pula os componentes (já) importados.

--update

Atualiza os componentes (já) importados.

Alterado na versão 2.9: Os parâmetros `--ignore` e `--update` existem para lidar com componentes já importados.

Exemplo de ficheiro JSON:

```
[
  {
    "slug": "po",
    "name": "Gettext PO",
    "file_format": "po",
    "filemask": "po/*.po",
    "new_lang": "none"
  },
  {
    "name": "Android",
    "filemask": "android/values-*/strings.xml",
    "template": "android/values/strings.xml",
    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]
```

Veja também:

import_memory

2.18.15 import_memory

weblate import_memory <file>

Novo na versão 2.20.

Importa um ficheiro TMX ou JSON para a memória de tradução do Weblate.

--language-map `LANGMAP`

Permite mapear idiomas no TMX para a memória de tradução do Weblate. Os códigos de idioma são mapeados após a normalização geralmente feita pelo Weblate.

`--language-map en_US:en` vai, por exemplo, importar todos os textos `en_US` como `en`.

Isso pode ser útil caso as localidades do seu ficheiro TMX não correspondam ao que usa no Weblate.

Veja também:

Memória de Tradução, *Esquema de memória de tradução Weblate*

2.18.16 import_project

weblate import_project <project> <gitrepo> <branch> <filemask>

Alterado na versão 3.0: O comando import_project agora é baseado na extensão *Descoberta de componentes*, levando a algumas mudanças no comportamento e quais parâmetros são aceitos.

Executa importação em lote de componentes para o projeto com base na máscara de ficheiros.

<project> é o nome de um projeto existente, para o qual os componentes devem ser importados.

O *<gitrepo>* define a URL do repositório Git a ser usada e *<branch>* significa o ramo Git. Para importar componentes de tradução adicionais de um componente Weblate existente, use uma URL *weblate://<project>/<component>* ao *<gitrepo>*.

O <*filemask*> define a descoberta de ficheiro para o repositório. Pode ser simplificado usando curingas ou pode usar todo o poder das expressões regulares.

A correspondência simples usa `**` para o nome do componente e `*` para o idioma, por exemplo: `**/*.po`

A expressão regular deve conter grupos chamados *component* e *language*. Por exemplo: `(?P<language>[^/]*)/(?P<component>[^-/]*)\.po`

A importação corresponde aos componentes existentes com base em ficheiros e adiciona aqueles que não existem. Isso não altera os já existentes.

--name-template TEMPLATE

Personaliza o nome de um componente usando a sintaxe do modelo Django.

Por exemplo: `Documentation: {{ component }}`

--base-file-template TEMPLATE

Personaliza o ficheiro base para traduções monolíngues.

Por exemplo: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Personaliza o ficheiro base para adicionar novas traduções.

Por exemplo: `{{ component }}/ts/en.ts`

--file-format FORMAT

Também pode especificar o formato de ficheiro a ser usado (veja *Formatos de ficheiros suportados*), a predefinição é a detecção automática.

--language-regex REGEX

Pode especificar a filtragem de idioma (veja *Configuração de componente*) com este parâmetro. Tem que ser uma expressão regular válida.

--main-component

Pode especificar qual componente será escolhido como o principal – aquele que realmente contém o repositório VCS.

--license NAME

Especifica a licença de tradução geral, projeto ou componente.

--license-url URL

Especifica a URL onde a licença de tradução encontra-se.

--vcs NAME

Caso precise especificar qual sistema de controle de versão usar, pode fazê-lo aqui. O controle de versão predefinido é Git.

Para dar alguns exemplos, vamos tentar importar dois projetos.

Primeiro as traduções do Manual do Debian, onde cada idioma separou uma pasta com as traduções de cada capítulo:

```
weblate import_project \
    debian-handbook \
    git://anonscm.debian.org/debian-handbook/debian-handbook.git \
    squeeze/master \
    '*/**.po'
```

Em seguida, a ferramenta Tanaguru, onde o formato do ficheiro precisa ser especificado, junto com o modelo de ficheiro base e como todos os componentes e traduções estão localizados numa única pasta:

```
weblate import_project \
    --file-format=properties \
    --base-file-template=web-app/tgol-web-app/src/main/resources/i18n/%s-I18N.
↪properties \
```

(continues on next page)

(continuação da página anterior)

```
    tanaguru \
    https://github.com/Tanaguru/Tanaguru \
    master \
    web-app/tgol-web-app/src/main/resources/i18n/**-I18N_*.properties
```

Um exemplo mais complexo de análise de nomes de ficheiro para obter o componente correto e o idioma de um nome de ficheiro como `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$'
```

Filtrando apenas traduções num idioma escolhido:

```
./manage import_project \
    --language-regex '^(cs|sk)$' \
    weblate \
    https://github.com/WeblateOrg/weblate.git \
    'weblate/locale/*/LC_MESSAGES/**.po'
```

Importando a documentação do Sphinx dividida em vários ficheiros:

```
$ weblate import_project --name-template 'Documentation: %s' \
    --file-format po \
    project https://github.com/project/docs.git master \
    'docs/locale/*/LC_MESSAGES/**.po'
```

A documentação de importação do Sphinx dividida em vários ficheiros e diretórios:

```
$ weblate import_project --name-template 'Directory 1: %s' \
    --file-format po \
    project https://github.com/project/docs.git master \
    'docs/locale/*/LC_MESSAGES/dir1/**.po'
$ weblate import_project --name-template 'Directory 2: %s' \
    --file-format po \
    project https://github.com/project/docs.git master \
    'docs/locale/*/LC_MESSAGES/dir2/**.po'
```

Veja também:

Exemplos mais detalhados encontram-se no capítulo starting, alternativamente quererá usar *import_json*.

2.18.17 importuserdata

weblate importuserdata <file.json>

Importa dados do utilizador de um ficheiro criado por *dumpuserdata*

2.18.18 importusers

weblate importusers --check <file.json>

Importa utilizadores do despejo JSON do banco de dados auth_users do Django.

--check

Com esta opção irá apenas verificar se determinado ficheiro pode ser importado e reportar possíveis conflitos decorrentes de nomes de utilizadores ou e-mails.

Pode descartar os utilizadores da instalação existente do Django usando:

```
weblate dumpdata auth.User > users.json
```

2.18.19 install_addon

Novo na versão 3.2.

weblate install_addon --addon ADDON <project|project/component>

Instala uma extensão para um conjunto de componentes.

--addon ADDON

Nome do complemento a ser instalado. Por exemplo, `weblate.gettext.customize`.

--configuration CONFIG

Configuração codificada em JSON de uma extensão.

--update

Atualiza a configuração existente da extensão.

Pode definir em qual projeto ou componente instalar a extensão (por exemplo, `weblate/application`) ou usar `--all` para incluir todos os componentes existentes.

Para instalar *Personalizar a saída gettext* para todos os componentes:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --
↪update --all
```

Veja também:

Extensões

2.18.20 list_languages

weblate list_languages <locale>

Lista os idiomas suportados na marcação MediaWiki – códigos de idioma, nomes em inglês e nomes localizados.

Isso é usado para gerar <https://wiki.l10n.cz/Slovn%C3%ADk_s_n%C3%A1zvy_jazyk%C5%AF>.

2.18.21 list_translators

weblate list_translators <project|project/component>

Lista tradutores por idioma contribuído ao projeto em questão:

```
[French]
Jean Dupont <jean.dupont@example.com>
[English]
John Doe <jd@example.com>
```

--language-code

Lista os nomes pelo código do idioma em vez do nome do idioma.

Pode definir qual projeto ou componente usar (por exemplo, `weblate/application`) ou usar `--all` para listar os tradutores de todos os componentes existentes.

2.18.22 list_versions

weblate list_versions

Lista todas as dependências do Weblate e as versões deles.

2.18.23 loadpo

weblate loadpo <project|project/component>

Recarrega traduções de disco (por exemplo, caso tenha feito algumas atualizações no repositório VCS).

--force

Força a atualização, mesmo que os ficheiros devam estar atualizados.

--lang LANGUAGE

Limita o processamento a um único idioma.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Nota: Raramente precisa invocar isso, o Weblate carregará automaticamente os ficheiros alterados para cada atualização do VCS. Isso é necessário no caso de alterar manualmente um repositório Weblate VCS subjacente ou em alguns casos especiais após uma atualização.

2.18.24 lock_translation

weblate lock_translation <project|project/component>

Impede outras traduções de um componente.

Dica: Útil caso queira fazer alguma manutenção no repositório subjacente.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Veja também:

unlock_translation

2.18.25 move_language

weblate move_language source target

Novo na versão 3.0.

Permite mesclar o conteúdo do idioma. Isso é útil ao atualizar para uma nova versão que contém apelidos para idiomas anteriormente desconhecidos que foram criados com o sufixo *(generated)*. Ele move todo o conteúdo do idioma fonte *source* para o de destino`target`.

Exemplo:

```
weblate move_language cze cs
```

Depois de mover o conteúdo, deve verificar se sobrou algo (isto está sujeito a condições de corrida quando alguém atualiza o repositório enquanto isso) e remover o idioma *(generated)*.

2.18.26 pushgit

weblate pushgit <project|project/component>

Faz um push das alterações com commit feito para o repositório VCS upstream.

--force-commit

Força o commit de todas as alterações pendentes, antes de fazer o push.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Nota: O Weblate faz o push das mudanças automaticamente se *Enviar ao submeter* na *Configuração de componente* está ativada, que é a predefinição.

2.18.27 unlock_translation

weblate unlock_translation <project|project/component>

Desbloqueia um determinado componente, disponibilizando-o para tradução.

Dica: Útil caso queira fazer alguma manutenção no repositório subjacente.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Veja também:

`lock_translation`

2.18.28 setupgroups

weblate setupgroups

Configura grupos predefinidos e, opcionalmente, atribui todos os utilizadores a esse grupo predefinido.

--no-privs-update

Desativa a atualização automática de grupos existentes (apenas adiciona novos).

--no-projects-update

Impede atualizações automáticas de grupos para projetos existentes. Isso permite adicionar grupos recém-adicionados a projetos existentes, consulte *Controlo de acesso ao projeto*.

Veja também:

Lista de privilégios e funções internas

2.18.29 setuplang

weblate setuplang

Atualiza a lista de idiomas definidos no Weblate.

--no-update

Desativa as atualizações automáticas de idiomas existentes (apenas adiciona novos).

2.18.30 updatechecks

weblate updatechecks <project|project/component>

Atualiza todas as verificações para todos os textos.

Dica: Útil para atualizações que fazem grandes mudanças nas verificações.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

2.18.31 updategit

weblate updategit <project|project/component>

Busca repositórios VCS remotos e atualiza o cache interno.

Pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Nota: Normalmente é melhor configurar os ganchos no repositório para acionar *Hooks de notificação*, ao invés da pesquisa regular por `updategit`.

2.19 Anúncios

Alterado na versão 4.0: Em versões anteriores, esse recurso era chamado de mensagens de quadro de comunicações.

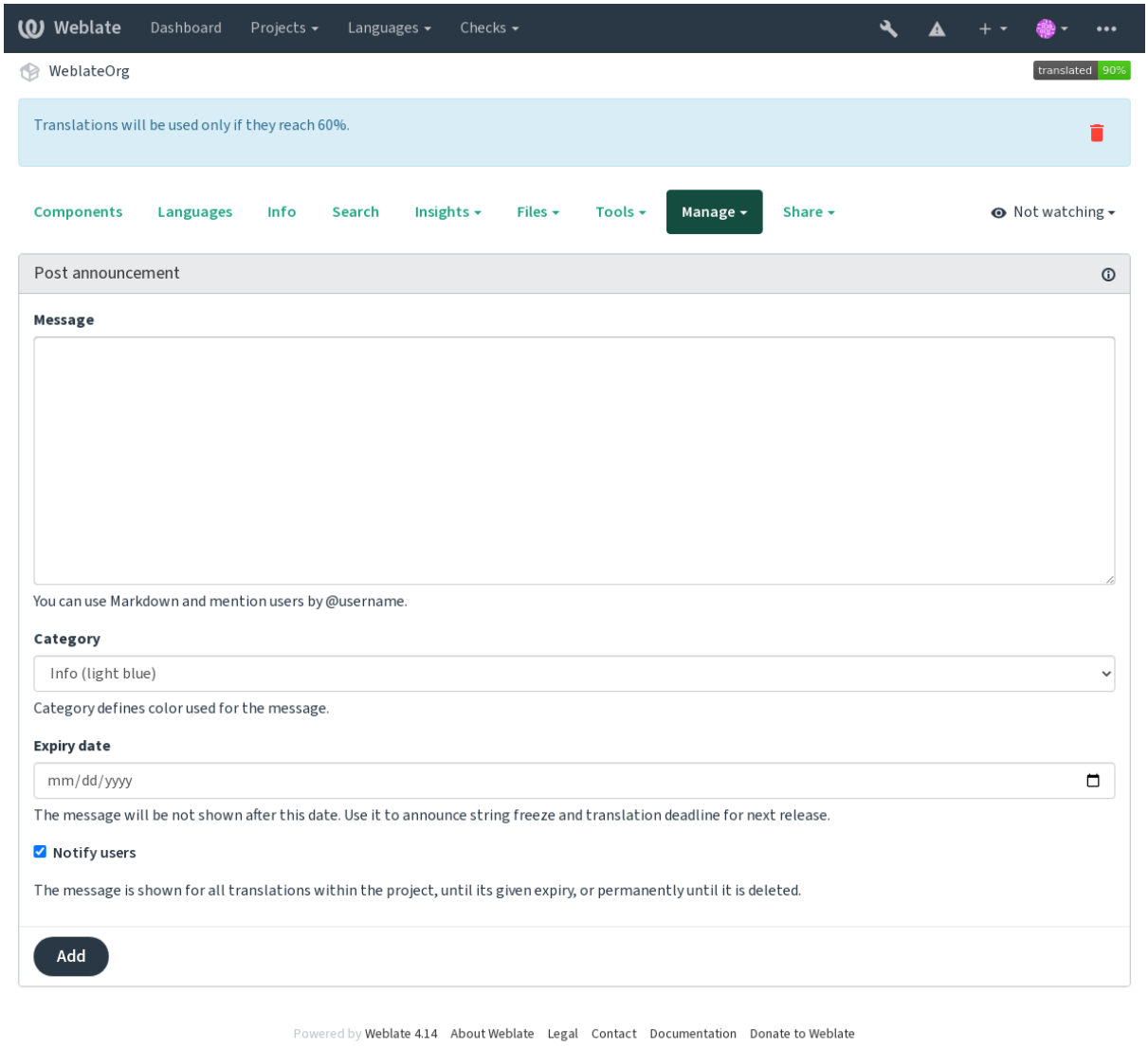
Forneça informações aos seus tradutores postando anúncios, em todo o site, por projeto, componente ou idioma.

Anuncie o propósito, prazos, estados ou especificar metas para tradução.

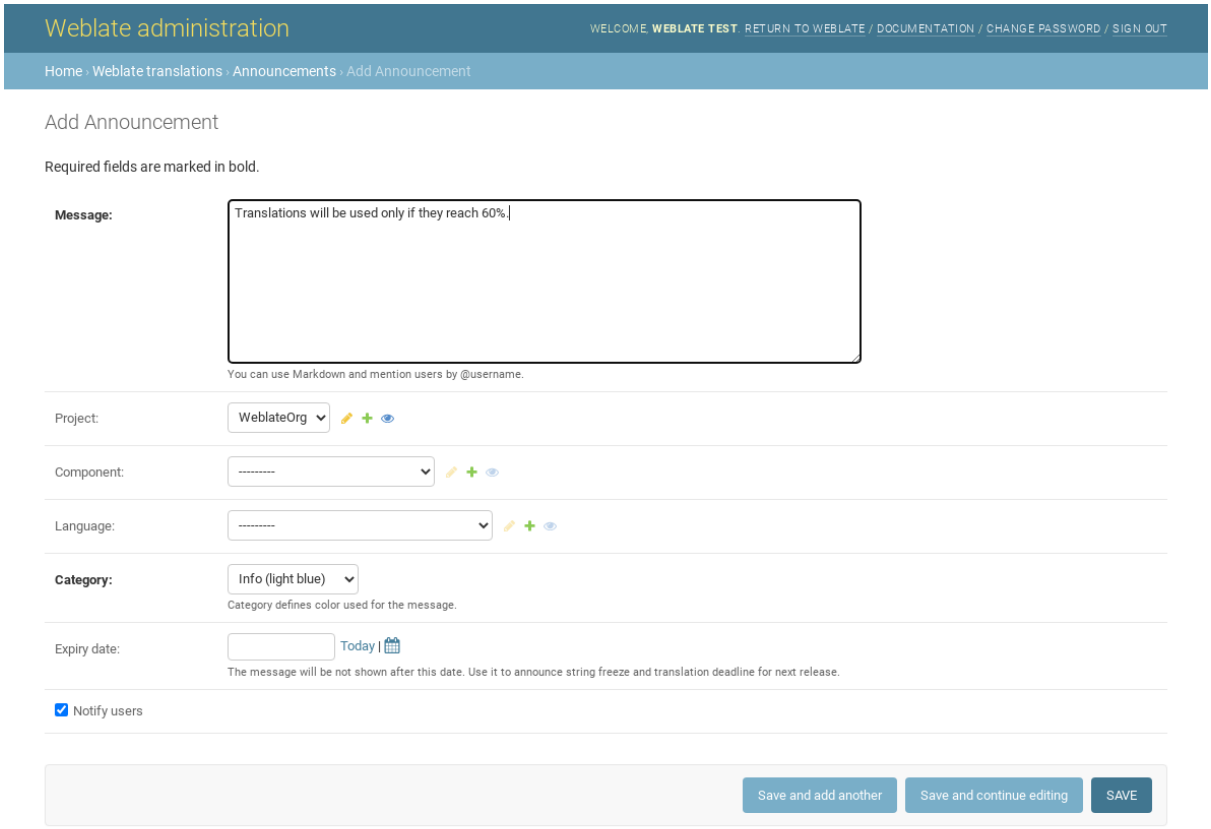
Os utilizadores receberão notificação sobre os anúncios de projetos assistidos (a menos que optem por não participar).

Isto pode ser útil para várias coisas, desde anunciar o propósito do site até especificar alvos para traduções.

Os anúncios podem ser publicados em cada nível no menu *Manage*, a usar :guilabel:"Publicar anúncio":



Ele também pode ser adicionado a usar a interface administrativa:



Os anúncios são então mostrados com base no seu contexto específico:

Nenhum contexto especificado

Mostrado no painel (página de chegada).

Projeto especificado

Mostrado dentro do projeto, incluindo todos os seus componentes e traduções.

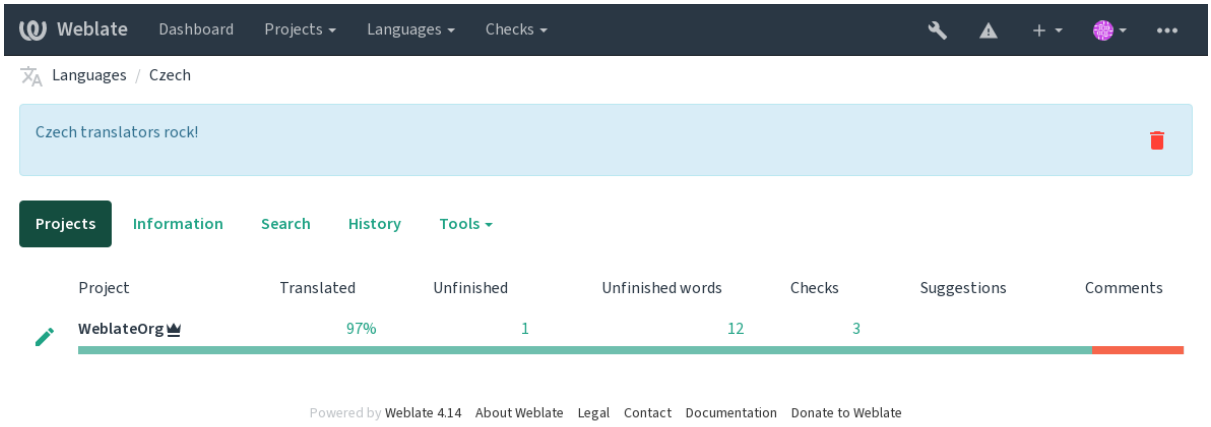
Componente especificado

Mostrado para um determinado componente e todas as traduções dele.

Idioma especificado

Mostrado na visão geral do idioma e todas as traduções nesse idioma.

Esta é a aparência na página de visão geral do idioma:



2.20 Lista de componentes

Especifique múltiplas listas de componentes para aparecer como opções no painel do utilizador, a partir do qual os utilizadores podem selecionar uma visualização como a visão predefinida. Veja *Painel* para saber mais.

Alterado na versão 2.20: Um estado vai ser apresentado para cada componente listado no painel.

Os nomes e conteúdos das listas de componentes podem ser especificados na interface administrativa, na secção *Component lists*. Cada lista de componentes deve ter um nome que é exibido ao utilizador e uma slug representando-a na URL.

Alterado na versão 2.13: Altera as configurações de painel para utilizadores anônimos da interface administrativa, a alterar qual painel é apresentado para utilizadores não autenticados.

2.20.1 Listas de componentes automáticas

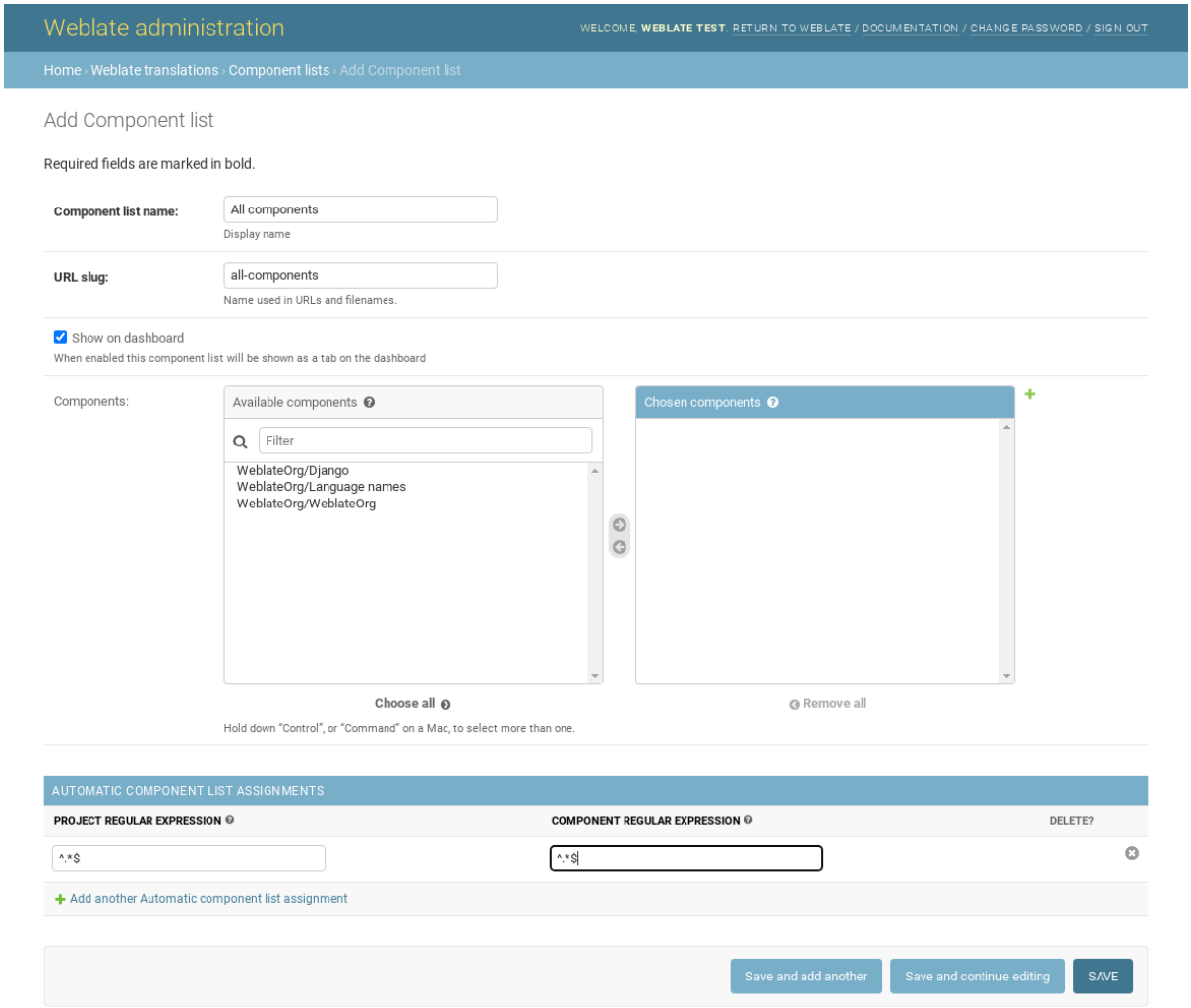
Novo na versão 2.13.

Adicione componentes à lista automaticamente com base nas suas slugs criando regras *Automatic component list assignment*.

- Útil para atualizar listas de componentes para grandes instalações, ou no caso de querer ter uma lista de componentes com todos os componentes na sua instalação do Weblate.

Dica: Faça uma lista de componentes contendo todos os componentes da sua instalação Weblate.

1. Define *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:



2.21 Módulos opcionais do Weblate

Vários módulos opcionais estão disponíveis para sua configuração.

2.21.1 Exportador git

Novo na versão 2.10.

Fornece acesso somente leitura ao repositório Git subjacente usando HTTP(S).

Instalação

1. Adicione `weblate.gitexport` às apps instaladas em `settings.py`:

```
INSTALLED_APPS += ("weblate.gitexport",)
```

2. Exporte repositórios existentes migrando seu banco de dados após a instalação:

```
weblate migrate
```

Uso

O módulo conecta-se automaticamente ao Weblate e define a URL do repositório exportado na *Configuração de componente*. Os repositórios são acessíveis na parte `/git/` da URL do Weblate, por exemplo, `https://example.org/git/weblate/main/`.

Repositórios para projetos disponíveis publicamente podem ser clonados sem autenticação:

```
git clone 'https://example.org/git/weblate/main/'
```

O acesso para navegar nos repositórios com acesso restrito (com *controle de acesso* *Privado* ou quando *`REQUIRE_LOGIN`* está ativado) requer um token de API que pode ser obtido no seu *perfil de utilizador*:

```
git clone 'https://user:KEY@example.org/git/weblate/main/'
```

Dica: Por predefinição, os membros ou o grupo *Utilizadores* e utilizador anônimo têm acesso aos repositórios para projetos públicos via *Acessar repositório* e funções de *Utilizador avançado*.

2.21.2 Faturação

Novo na versão 2.4.

Isso é usado no Hosted Weblate para definir planos de cobrança, rastrear faturas e limites de uso.

Instalação

1. Add `weblate.billing` to installed apps in `settings.py`:

```
INSTALLED_APPS += ("weblate.billing",)
```

2. Execute a migração do banco de dados para instalar opcionalmente estruturas de banco de dados adicionais para o módulo:

```
weblate migrate
```

Uso

Após a instalação, pode controlar a cobrança na interface de administração. Os utilizadores com cobrança ativada obterão a nova guia *Cobrança* no *Perfil do utilizador* deles.

O módulo de cobrança também permite que os administradores do projeto criem novos projetos e componentes sem serem superutilizadores (veja add-projects). Isso é possível quando as seguintes condições são atendidas:

- A cobrança está nos limites configurados dele (qualquer uso excessivo resulta no bloqueio da criação do projeto/componente) e pago (se o preço dele for diferente de zero)
- O utilizador é administrador do projeto existente com cobrança ou o utilizador é proprietário da cobrança (este último é necessário ao criar uma nova cobrança para que os utilizadores possam importar novos projetos).

Após a criação do projeto, o utilizador pode escolher qual faturamento deve ser cobrado pelo projeto, caso tenha acesso a mais deles.

2.21.3 Legal

Novo na versão 2.15.

Isso é usado em Weblate hospedado para fornecer documentos legais necessários. Ele vem fornecido com documentos em branco e espera-se que preencha os seguintes modelos nos documentos:

`legal/documents/tos.html`
Documento de termos de serviço

`legal/documents/privacy.html`
Documento de política de privacidade

`legal/documents/summary.html`
Visão geral breve dos termos de serviço e política de privacidade

---

Nota: Documentos legais para o serviço Hosted Weblate estão disponíveis neste repositório Git <https://github.com/WeblateOrg/wllegal/tree/main/wllegal/templates/legal/documents>.

Muito provavelmente, eles não serão úteis diretamente para si, mas podem ser úteis como um ponto de partida se ajustados para atender às suas necessidades.

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. Execute a migração do banco de dados para instalar opcionalmente estruturas de banco de dados adicionais para o módulo:

```
weblate migrate
```

3. Edite os documentos jurídicos na pasta `weblate/legal/templates/legal/` para corresponder ao seu serviço.

Uso

Após a instalação e edição, os documentos legais são exibidos na interface de utilizador do Weblate.

2.21.4 Avatares

Os avatares são descarredaos e armazenados em cache no lado do servidor para reduzir o vazamento de informações para os sites que os servem por predefinição. O suporte embutido para buscar avatares de endereços de e-mail configurados para isso pode ser desligado usando `ENABLE_AVATARS`.

Atualmente, o Weblate oferece suporte a:

- Gravatar
- Libravatar

Veja também:

Cache de avatares, `AVATAR_URL_PREFIX`, `ENABLE_AVATARS`

2.21.5 Proteção contra spam

Pode se proteger contra spam por utilizadores usando o serviço Akismet.

1. Instale o módulo Python *akismet* (ele já está incluído na imagem Docker oficial).
2. Obtenha a chave de API do Akismet.
3. Armazene-a como `AKISMET_API_KEY` ou `WEBLATE_AKISMET_API_KEY` no Docker.

O seguinte conteúdo é enviado ao Akismet para verificação:

- Sugestões de utilizador não autenticados
- Descrições e ligações de projetos e componentes

Nota: Este (entre outras coisas) depende do endereço IP do cliente, veja *Executar por trás de um proxy reverso* para configurá-lo apropriadamente.

Veja também:

Executar por trás de um proxy reverso, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Assinando commits do Git com GnuPG

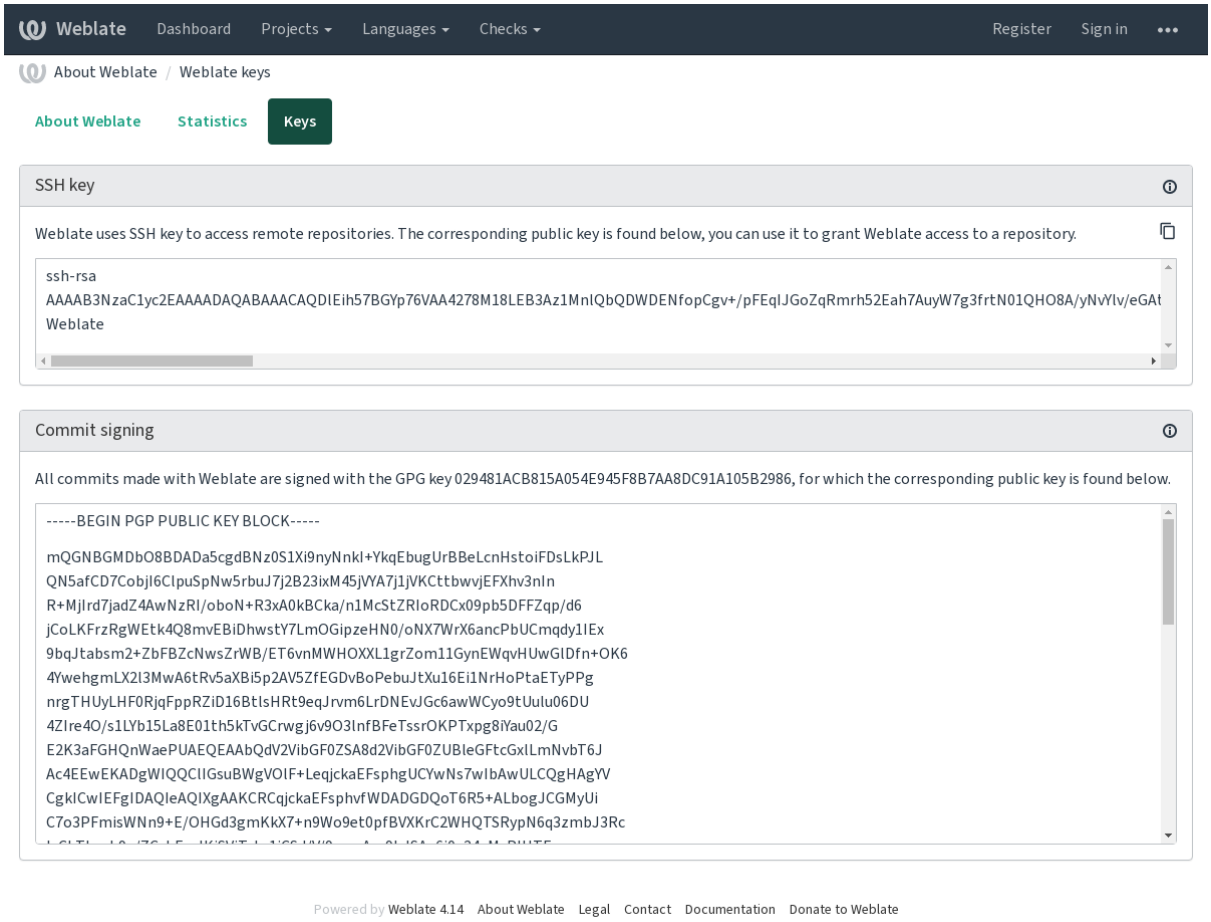
Novo na versão 3.1.

Todos os commits podem ser assinados pela chave GnuPG da instância Weblate.

1. Turn on `WEBLATE_GPG_IDENTITY`. (Weblate will generate a GnuPG key when needed and will use it to sign all translation commits.)

Este recurso precisa do GnuPG 2.1 ou mais recente instalado.

Pode encontrar a chave em `DATA_DIR` e a chave pública é mostrada na página «Sobre»:



2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking gpg.

Veja também:

WEBLATE_GPG_IDENTITY

2.21.7 Limitação de taxa

Alterado na versão 3.2: A limitação de taxa agora aceita uma configuração mais refinada.

Alterado na versão 4.6: A limitação de taxa não se aplica mais a superutilizadores.

Várias operações no Weblate são limitadas por taxas. No máximo *RATELIMIT_ATTEMPTS* tentativas são permitidas dentro de *RATELIMIT_WINDOW* segundos. O utilizador é bloqueado por *RATELIMIT_LOCKOUT*. Há também configurações específicas para escopos como, por exemplo, `RATELIMIT_CONTACT_ATTEMPTS` ou `RATELIMIT_TRANSLATE_ATTEMPTS`. A tabela abaixo é uma lista completa de escopos disponíveis.

As seguintes operações estão sujeitas a limitação de taxa:

Nome	Âmbito	Tentativas permitidas	Janela de limite de tempo	Período de bloqueio
Registo	`REGISTRATION`	5	300	600
Enviando mensagem para administradores	`MESSAGE`	5	300	600
Autenticação de palavra-passe ao entrar	`LOGIN`	5	300	600
Pesquisa em todo o site	`SEARCH`	6	60	60
Traduzir	`TRANSLATE`	30	60	600
Adicionando ao glossário	`GLOSSARY`	30	60	600
Iniciando a tradução para um novo idioma	`LANGUAGE`	2	300	600
Criando um novo projeto	`PROJECT`	5	600	600

If a user fails to sign in *`AUTH_LOCK_ATTEMPTS`* times, password authentication will be turned off on the account until having gone through the process of having its password reset.

As configurações também podem ser aplicadas no contentor do Docker adicionando o prefixo `WEBLATE_` ao nome da configuração, por exemplo *`RATELIMIT_ATTEMPTS`* torna-se *`WEBLATE_RATELIMIT_ATTEMPTS`*.

A API possui configurações separadas de limitação de taxa, consulte *Limitação de taxa da API*.

Veja também:

Limitação de taxa, *Executar por trás de um proxy reverso*, *Limitação de taxa da API*

2.21.8 Integração com Fedora Messaging

Fedora Messaging é um editor baseado em AMQP para todas as mudanças que acontecem no Weblate. Pode conectar serviços adicionais nas mudanças que acontecem no Weblate usando isso.

A integração com Fedora Messaging está disponível como um módulo Python separado `weblate-fedora-messaging`. Consulte <https://github.com/WeblateOrg/fedora_messaging/> para obter instruções de configuração.

2.22 Personalizar o Weblate

Amplie e personalize a usar Django e Python. Contribua as suas alterações para o upstream acima para que todos possam se beneficiar. Isso reduz os seus custos de manutenção; código no Weblate é cuidado ao alterar interfaces internas ou refatorar o código.

> **Aviso:** Nem interfaces internas nem modelos são considerados uma API estável. Por favor, revise as suas próprias personalizações para cada atualização, as interfaces ou a semântica deles podem mudar sem aviso prévio.

Veja também:

Contribuir ao Weblate

2.22.1 Criar um módulo Python

Se não conheçe o Python, pode olhar para Python For Beginners, que explica o básico e aponta aos tutoriais adicionais.

Para escrever algum código Python personalizado (chamado de módulo), é necessário um lugar para armazená-lo, seja no caminho do sistema (geralmente algo como `/usr/lib/python3.9/site-packages/`) ou no diretório Weblate, que também é adicionado ao caminho de pesquisa do interpretador.

Melhor ainda, transforme a sua personalização num pacote Python adequado:

1. Crie uma pasta para o seu pacote (usaremos *weblate_customization*).
2. Dentro dele, crie um ficheiro `setup.py` para descrever o pacote:

```
from setuptools import setup

setup(
    name="weblate_customization",
    version="0.0.1",
    author="Your name",
    author_email="yourname@example.com",
    description="Sample Custom check for Weblate.",
    license="GPLv3+",
    keywords="Weblate check example",
    packages=["weblate_customization"],
)
```

3. Crie uma pasta para o módulo Python (também chamado de `weblate_customization`) para o código de personalização.
4. Dentro dele, crie um ficheiro `__init__.py` para garantir que o Python possa importar o módulo.
5. Este pacote agora pode ser instalado a usar **`pip install -e`**. Mais informações a serem encontradas em Editable installs.
6. Uma vez instalado, o módulo pode ser usado na configuração Weblate (por exemplo, `weblate_customization.checks.FooCheck`).

Sua estrutura de módulo deve ser assim:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

Pode encontrar um exemplo de personalização do Weblate em <https://github.com/WeblateOrg/customize-example>, ele abrange todos os tópicos descritos abaixo.

2.22.2 Alterar o logotipo

1. Crie uma app Django simples a conter os ficheiros estáticos que deseja substituir (veja *Criar um módulo Python*).

   A marca aparece nos ficheiros seguintes:

   `icons/weblate.svg`
   Logotipo mostrado na barra de navegação.

   `logo-*.png`
   Ícones web dependendo da resolução do ecrã e do navegador web.

   `favicon.ico`
   Ícone web usado por navegadores legados.

weblate-*.png
Avatares para bots ou utilizadores 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 ficheiros estáticos servidos aos clientes.

Veja também:

How to manage static files (e.g. images, JavaScript, CSS), *Servir ficheiros estáticos*

2.22.3 Verificações de qualidade personalizadas, extensões e correções automáticas

Para instalar o seu código para *Correções automáticas personalizadas*, *Escrever próprias verificações* ou *Escrevendo extensões* no Weblate:

1. Ponha os ficheiros no seu módulo Python a conter a personalização ao Weblate (veja *Criar um módulo Python*).
2. Adicione o caminho totalmente qualificado dele à classe Python nas configurações dedicadas (*WEBLATE_ADDONS*, *CHECK_LIST* ou *AUTOFIX_LIST*):

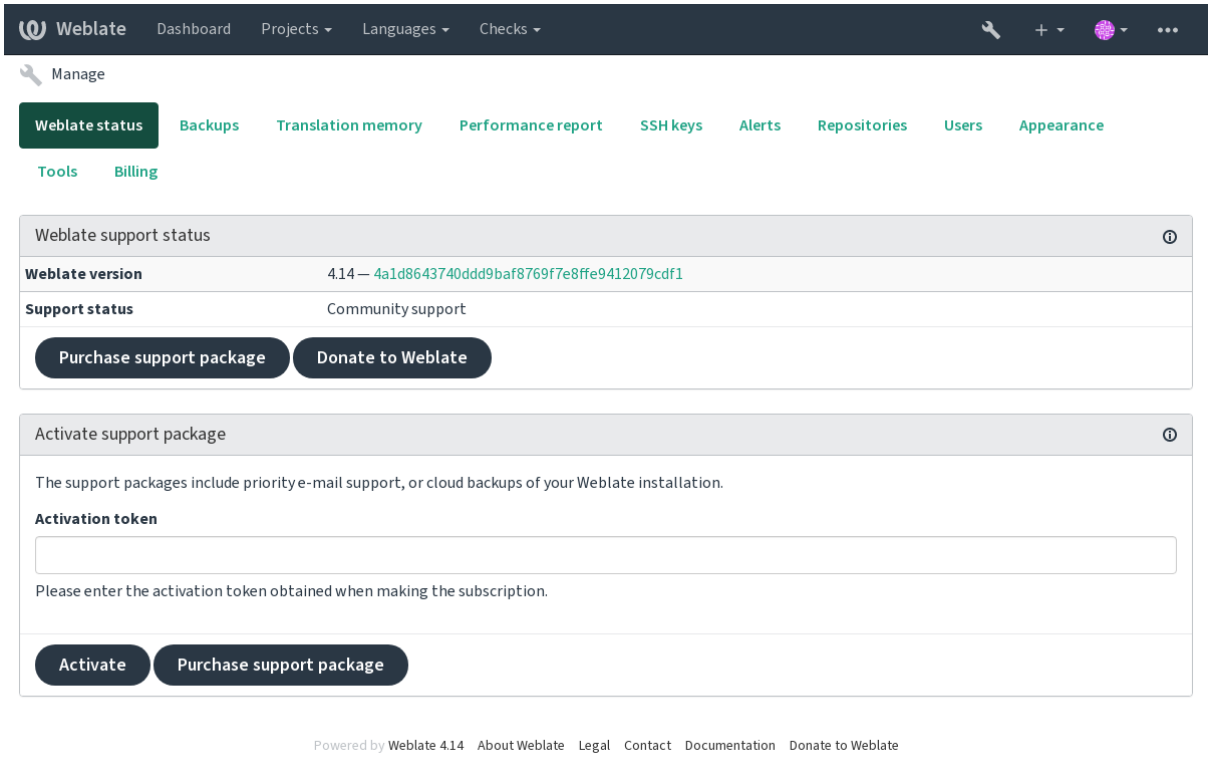
```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Veja também:

Correções automáticas personalizadas, *Escrever próprias verificações*, *Escrevendo extensões*, *Escrevendo scripts para extensões*

2.23 Interface de gestão

A interface de gestão oferece configurações de administração sob a URL `/manage/`. Está disponível para utilizadores que se inscrevem com privilégios administrativos, acessíveis a usar o ícone da chave inglesa no canto superior direito:



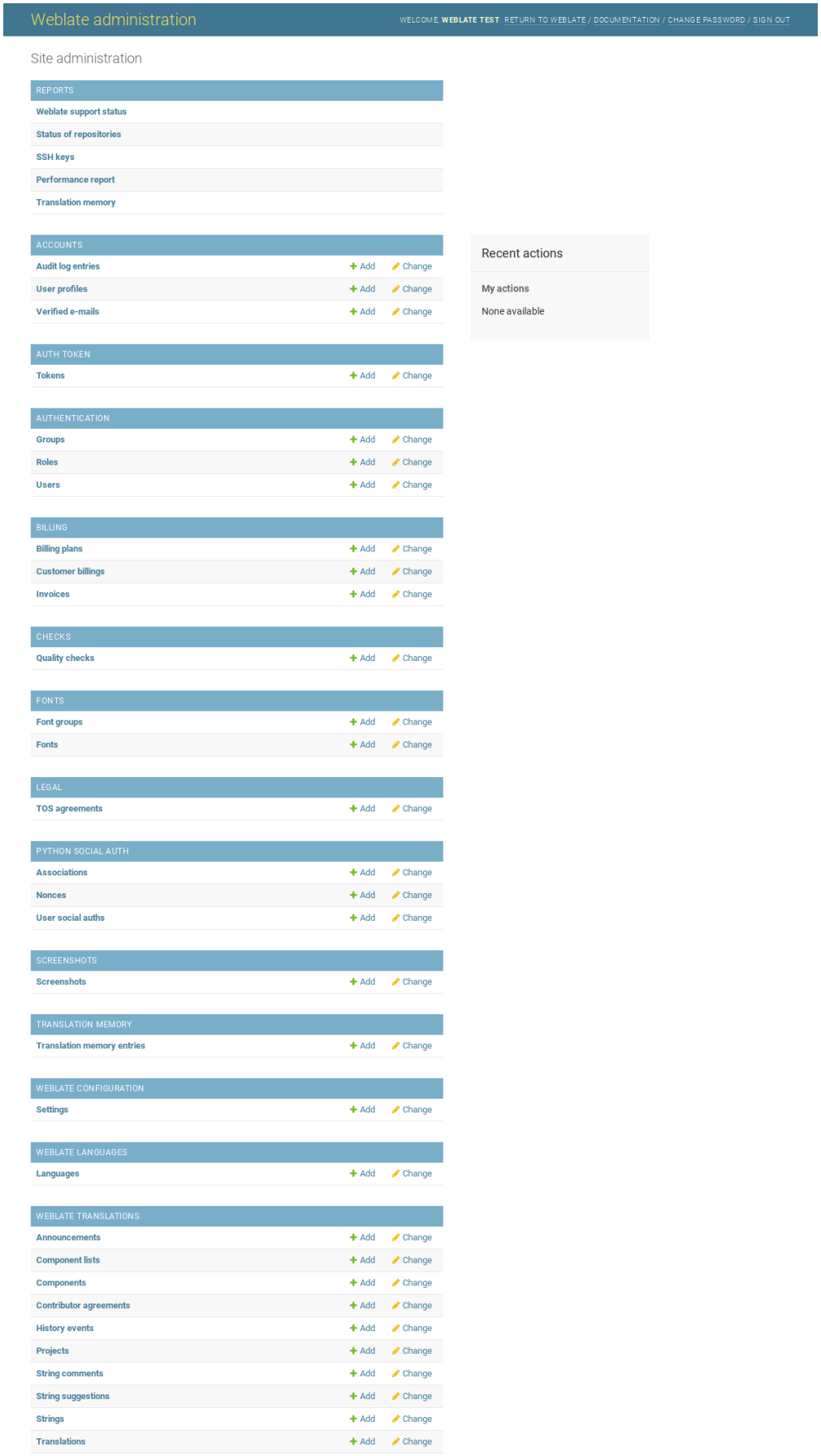
Ela inclui uma visão geral básica do seu Weblate:

- Estado de suporte, veja *Obter suporte para o Weblate*
- Backups, veja *Fazer backup e mover o Weblate*
- Memória de tradução compartilhada, veja *Memória de Tradução*
- Relatório de desempenho para revisar a saúde do Weblate e a duração das filas do Celery
- Gestão de chaves SSH, veja *Repositórios SSH*
- Visão geral de alertas para todos os componentes, veja alerts

2.23.1 A interface administrativa do Django

> **Aviso:** Será removido no futuro, pois o uso dele é desencorajado — a maioria das funcionalidades pode ser geridas diretamente no Weblate.

Aqui pode gerir objetos armazenados no banco de dados, tais como utilizadores, traduções e outras configurações:


Weblate administration
WELCOME, WEBLATE TEST. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Site administration
REPORTS
Weblate support status
Status of repositories
SSH keys
Performance report
Translation memory
ACCOUNTS
Audit log entries + Add Change
User profiles + Add Change
Verified e-mails + Add Change
AUTH TOKEN
Tokens + Add Change
AUTHENTICATION
Groups + Add Change
Roles + Add Change
Users + Add Change
BILLING
Billing plans + Add Change
Customer billings + Add Change
Invoices + Add Change
CHECKS
Quality checks + Add Change
FONTS
Font groups + Add Change
Fonts + Add Change
LEGAL
TOS agreements + Add Change
PYTHON SOCIAL AUTH
Associations + Add Change
Nonces + Add Change
User social auths + Add Change
SCREENSHOTS
Screenshots + Add Change
TRANSLATION MEMORY
Translation memory entries + Add Change
WEBLATE CONFIGURATION
Settings + Add Change
WEBLATE LANGUAGES
Languages + Add Change
WEBLATE TRANSLATIONS
Announcements + Add Change
Component lists + Add Change
Components + Add Change
Contributor agreements + Add Change
History events + Add Change
Projects + Add Change
String comments + Add Change
String suggestions + Add Change
Strings + Add Change
Translations + Add Change
Recent actions
My actions
None available

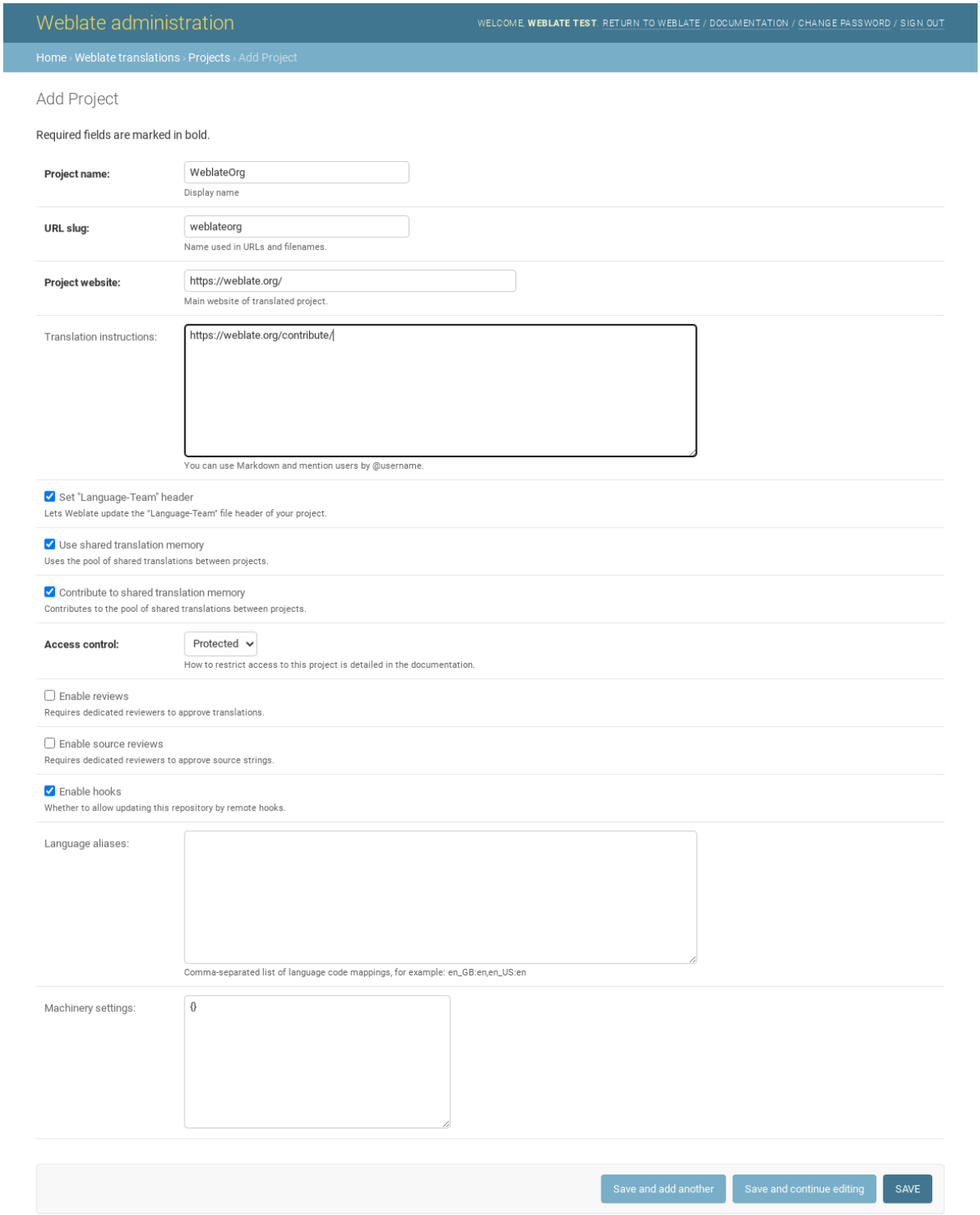
Na secção *Relatórios* pode verificar o estado do seu site, ajustá-lo para produção ou gerir chaves SSH usadas para acessar *Acessando repositórios*.

Gerir objetos de banco de dados em qualquer uma das secções abaixo. A mais interessante é provavelmente *Traduções do Weblate*, onde pode gerir projetos traduzíveis, veja *Configuração de projeto* e *Configuração de componente*.

Idiomas do Weblate detém as definições de idiomas, explicado melhor em *Definições de idioma*.

Adicionar um projeto

A adição de um projeto serve como contentor para todos os componentes. Normalmente cria um projeto para um software, ou livro (Veja *Configuração de projeto* para informações sobre parâmetros individuais):



Veja também:

Configuração de projeto

Componentes bilíngues

Uma vez que adicionou um projeto, os componentes de tradução podem ser adicionados-lo. (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

Weblate administration
WELCOME, WEBLATE TEST. REPORTS TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Home › Weblate translations › Components › Add Component
Add Component
IMPORT SPEED DOCUMENTATION
Required fields are marked in bold.
Component name:
Language names
Display name
URL slug:
language-names
Name used in URLs and filenames.
Project:
WeblateOrg
Version control system:
Git
Version control system to use to access your repository containing translations. You can also choose additional integration with third party providers to submit merge requests.
Source code repository:
https://github.com/WeblateOrg/demo.git
URL of a repository, use weblate://project/component to share it with other component.
Repository push URL:
URL of a push repository, pushing is turned off if empty.
Repository browser:
https://github.com/WeblateOrg/demo/blob/{{branch}}/{{filename}}#L{
Link to repository browser, use {{branch}} for branch, {{filename}} and {{pos}} as filename and line placeholders. You might want to strip leading directory by using {{filenameparent0}}.
Exported repository URL:
URL of repository where users can fetch changes from Weblate
Source string bug reporting address:
E-mail address for reports on errors in source strings. Leave empty for no e-mails.
Repository branch:
Repository branch to translate
Push branch:
Branch for pushing changes, leave empty to use repository branch
File mask:
weblate/langdata/locale/*/LC_MESSAGES/djo
Path of files to translate relative to repository root, use * instead of language code, for example po/* 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.
File format:
gettext PO file
Locked
Locked component will not get any translation updates.
Allow translation propagation
Whether translation updates in other components will cause automatic translation in this one
Turn on suggestions
Whether to allow translation suggestions at all.
Suggestion voting
Users can only vote for suggestions and can't make direct translations.
Autoaccept suggestions:
0
Automatically accept suggestions with this number of votes, use 0 to turn it off.
Translation flags:
Additional comma-separated flags to influence Weblate behavior.
Enforced checks:
[]
List of checks which can not be ignored.
Translation license:
GNU General Public License v3.0 or later
Contributor agreement:
User agreement which needs to be approved before a user can translate this component.
Adding new translation:
Create new language file
How to handle requests for creating new translations.
Language code style:
Default based on the file format
Customize language code used to generate the filename for translations created by Weblate.
Manage strings
Enables adding and removing strings straight from Weblate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.
Merge style:
Rebase
Define whether Weblate should merge the upstream repository or rebase changes onto it.
Commit message when translating:
Translated using Weblate ({{ language_name }})
Currently translated at {{ stats.translated_percent }}% ({{ stats.translated }} of {{ stats.all }} strings)
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Commit message when adding translation:
Added translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when removing translation:
Deleted translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when merging translation:
Merge branch '{{ component_remote_branch }}' into Weblate.
You can use template language for various info, please consult the documentation for more details.
Commit message when add-on makes a change:
Update translation files
Updated by "{{ addon_name }}" hook in Weblate.
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Merge request message:
Translations update from {{ site_title }}
Translations update from {{ site_title }}({{ site_url }}) for {{ project_name }}/{{ component_name }}({{{url}}}.
{%, if component_linked_childs %}
It also includes following components:
{% for linked in component_linked_childs %}
* {{ linked.project_name }}/{{ linked.name }}({{ linked.url }})
{% endfor %}
{% endif %}
You can use template language for various info, please consult the documentation for more details.
Push on commit
Whether the repository should be pushed upstream on every commit.
Age of changes to commit:
24
Time in hours after which any pending changes will be committed to the VCS.
Lock on error
Whether the component should be locked on repository errors.
Source language:
English
Language used for source strings in all components.
Language filter:
^(nl|fr|hu)$
Regular expression used to filter translation files when scanning for file mask.
Variants regular expression:
Regular expression used to determine variants of a string.
Priority:
Medium
Components with higher priority are offered first to translators.
Restricted component
Restrict access to the component to only those explicitly given permission.
Share in projects:
WeblateOrg
Choose additional projects where this component will be listed. Hold down "Control", or "Command" on a Mac, to select more than one.
Use as a glossary
Glossary color:
Silver
Remote revision:
Local revision:
Save and add another
Save and continue editing
SAVE

Veja também:

Configuração de componente, *Formatos bilíngues e monolíngues*

Componentes monolínguas

Para facilitar a tradução destes, forneça um ficheiro de modelo contendo o mapeamento de IDs de mensagem para respectivo idioma fonte dele (geralmente inglês). (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

Weblate administration
WELCOME, WEBLATE TEST. RETURN TO WEBLATE / DOCUMENTATION / CHANGE PASSWORD / SIGN OUT
Home › Weblate translations › Components › Add Component
Add Component
IMPORT SPHINX DOCUMENTATION
Required fields are marked in bold.
Component name:
Display name
URL slug:
Name used in URLs and filenames.
Project:
Version control system:
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:
URL of a repository use weblate://project/component to share it with other component.
Repository push URL:
URL of a push repository, pushing is turned off if empty.
Repository browser:
Link to repository browser, use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders. You might want to strip leading directory by using {{filename:sequence:dir}}.
Exported repository URL:
URL of repository where users can fetch changes from Weblate.
Source string bug reporting address:
E-mail address for reports on errors in source strings. Leave empty for no e-mails.
Repository branch:
Repository branch to translate.
Push branch:
Branch for pushing changes, leave empty to use repository branch.
File mask:
Path of files to translate relative to repository root, use * instead of language code, for example: po/*.po or locale/*/LC_MESSAGES/djangojs.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:
Filename of file used for creating new translations. For gettext choose .pot file.
File format:
Locked
Locked component will not get any translation updates.
Allow translation propagation
Whether translation updates in other components will cause automatic translation in this one.
Turn on suggestions
Whether to allow translation suggestions at all.
Suggestion voting
Users can only vote for suggestions and can't make direct translations.
Autoaccept suggestions:
Automatically accept suggestions with this number of votes, use 0 to turn it off.
Translation flags:
Additional comma-separated flags to influence Weblate behavior.
Enforced checks:
List of checks which can not be ignored.
Translation license:
MIT License
Contributor agreement:
User agreement which needs to be approved before a user can translate this component.
Adding new translation:
Create new language file
How to handle requests for creating new translations.
Language code style:
Default based on the file format
Customize language code used to generate the filename for translations created by Weblate.
Manage strings
Enables adding and removing strings straight from Weblate. If your strings are extracted from the source code or managed externally you probably want to keep it disabled.
Merge style:
Rebase
Define whether Weblate should merge the upstream repository or rebase changes onto it.
Commit message when translating:
Translated using Weblate ({{ language_name }})
Currently translated at {{ stats.translated_percent }}% ({{ stats.translated }} of {{ stats.all }} strings)
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Commit message when adding translation:
Added translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when removing translation:
Deleted translation using Weblate ({{ language_name }})
You can use template language for various info, please consult the documentation for more details.
Commit message when merging translation:
Merge branch '{{ component_remote_branch }}' into Weblate.
You can use template language for various info, please consult the documentation for more details.
Commit message when add-on makes a change:
Update translation files
Updated by "{{ addon_name }}" hook in Weblate.
Translation: {{ project_name }}/{{ component_name }}
Translate-URL: {{ url }}
You can use template language for various info, please consult the documentation for more details.
Merge request message:
Translations update from {{ site_title }}
Translations update from {{ site_title }}({{ site_url }}) for {{ project_name }}/{{ component_name }}({{ url }}).
{% if component_linked_childs %}
It also includes following components:
{% for linked in component_linked_childs %}
* {{ linked.project_name }}/{{ linked.name }}({{ linked.url }})
{% endfor %}
{% endif %}
You can use template language for various info, please consult the documentation for more details.
Push on commit
Whether the repository should be pushed upstream on every commit.
Age of changes to commit:
24
Time in hours after which any pending changes will be committed to the VCS.
Lock on error
Whether the component should be locked on repository errors.
Source language:
English
Language used for source strings in all components.
Language filter:
^[^,]+$
Regular expression used to filter translation files when scanning for file mask.
Variants regular expression:
Regular expression used to determine variants of a string.
Priority:
Medium
Components with higher priority are offered first to translators.
Restricted component
Restrict access to the component to only those explicitly given permission.
Share in projects:
WeblateOrg
Choose additional projects where this component will be listed. Hold down "Control", or "Command" on a Mac, to select more than one.
Use as a glossary
Glossary color:
Silver
Remote revision:
Local revision:
Save and add another
Save and continue editing
SAVE

Veja também:

Configuração de componente, *Formatos bilíngues e monolíngues*

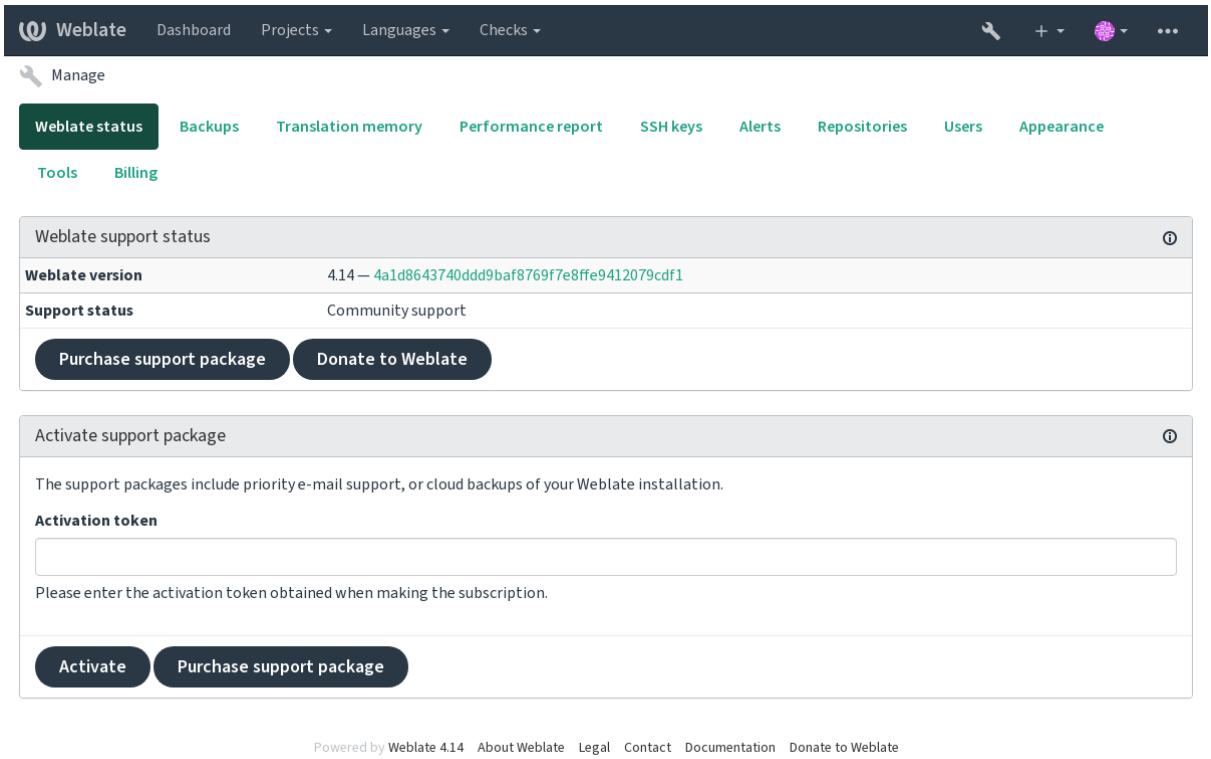
2.24 Obter suporte para o Weblate

Weblate é um software livre protegido por copyleft e com apoio comunitário. Os assinantes recebem apoio prioritário sem custo adicional. Pacotes de ajuda pré-pago estão disponíveis para todos. Pode encontrar mais informações sobre as ofertas de apoio atuais em <https://weblate.org/support/>.

2.24.1 Integrando o apoio

Novo na versão 3.8.

Os pacotes de apoio adquiridos podem ser integrados opcionalmente à sua gestão de assinatura do Weblate, de onde encontrará uma ligação a ele. Detalhes básicos da instância sobre a sua instalação também são relatados de volta ao Weblate desta forma.



2.24.2 Dados enviados ao Weblate

- URL onde a sua instância do Weblate está configurada
- Título do seu site
- A versão do Weblate que está a executar
- Contagem de alguns objetos no seu banco de dados Weblate (projetos, componentes, idiomas, cadeias fonte e utilizadores)
- A chave pública SSH da sua instância

Além disso, quando *Descubra Weblate* está ativado:

- Lista de projetos públicos (nome, URL e site)

Nenhum outro dado é enviado.

2.24.3 Serviços de integração

- Veja se o seu pacote de apoio ainda é válido
- *Armazenamento de backup provisionado do Weblate*
- *Descubra Weblate*

Dica: Os pacotes de apoio adquiridos já estão ativados no momento da compra e podem ser usados sem integrá-los.

2.24.4 Descubra Weblate

Novo na versão 4.5.2.

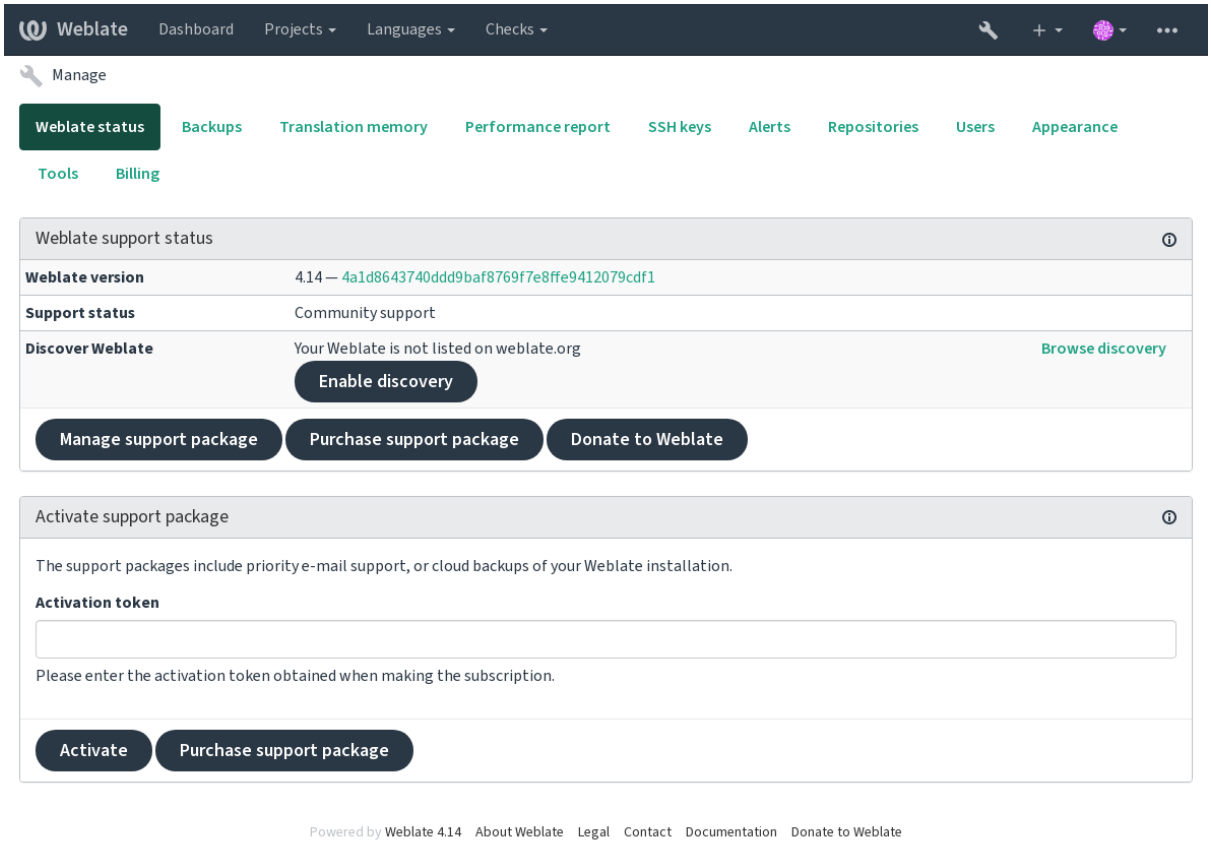
Nota: Este recurso está atualmente num beta inicial.

Descubra Weblate é um serviço opcional que facilita encontrar comunidades e servidores Weblate. Os utilizadores podem navegar por serviços cadastrados em <https://weblate.org/discover/> e lá encontrar projetos para contribuir.

Como fazer para ser listado

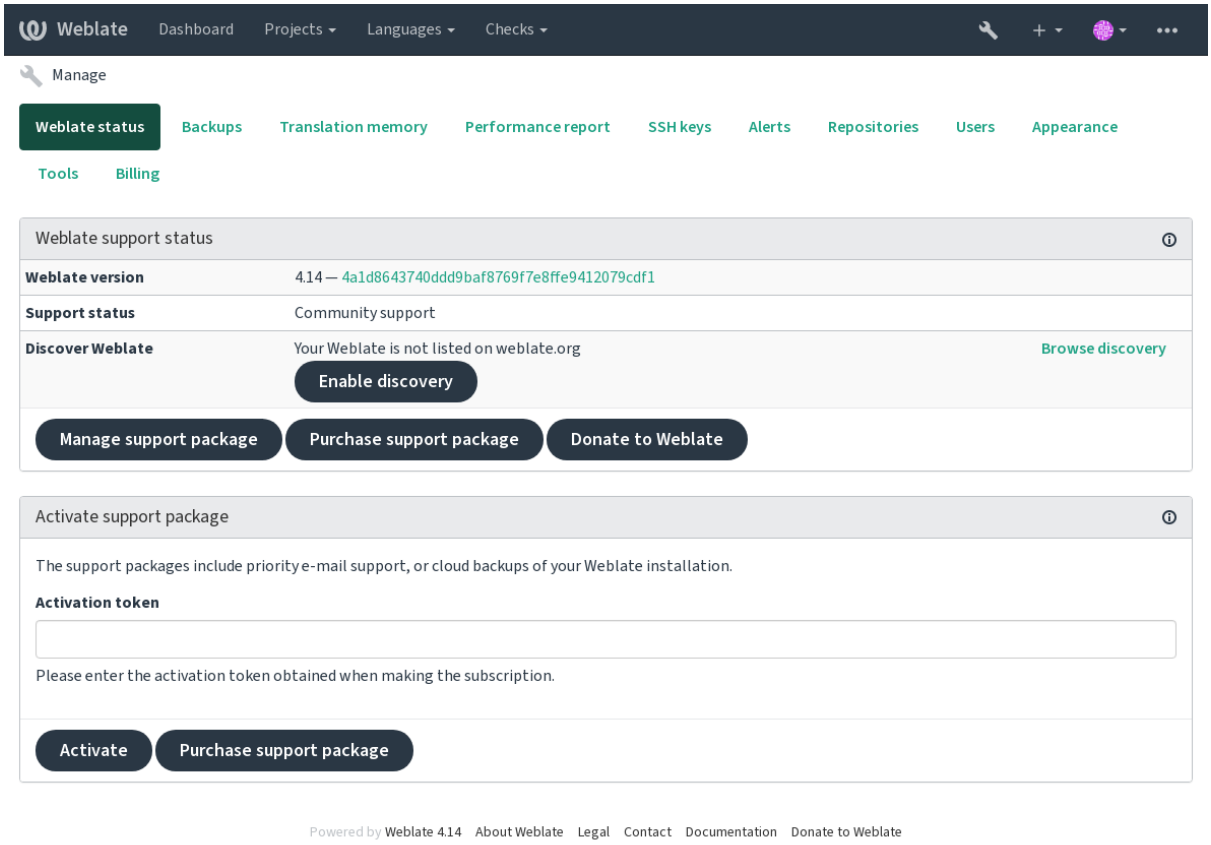
Dica: Participar do Descubra Weblate faz com que o Weblate envie algumas informações sobre o seu servidor, veja *Dados enviados ao Weblate*.

Para listar o seu servidor com uma assinatura de suporte ativa (veja *Integrando o apoio*) no Descubra Weblate tudo que precisa fazer é ativar isso no painel de gestão:



Listando seu servidor sem uma assinatura de suporte no Descubra Weblate:

1. Registe-se em <https://weblate.org/user/>
2. Registe o seu servidor Weblate no banco de dados de descoberta em <https://weblate.org/subscription/discovery/>
3. Confirme a ativação do serviço no seu Weblate e ative a lista de descoberta na sua página de gestão do Weblate usando o botão *Ativar descoberta*:



Personalizando a listagem

Pode personalizar a lista fornecendo um texto e uma imagem (570 x 260 pixels) em <https://weblate.org/user/>.

2.25 Documentos legais

Nota: Aqui encontrará várias informações legais que pode precisar para operar Weblate em certas jurisdições legais. É fornecido como um meio de orientação, sem qualquer garantia de precisão ou correção. Em última análise, é a sua responsabilidade de garantir que o seu uso do Weblate esteja em conformidade com todas as leis e regulamentos aplicáveis.

2.25.1 ITAR e outros controles de exportação

O Weblate pode ser usado dentro do seu próprio datacenter ou nuvem privada virtual. Como tal, ele pode ser usado para armazenar informações ITAR ou outras controladas por exportação; no entanto, os utilizadores finais são responsáveis por garantir tal conformidade.

O serviço Hosted Weblate não foi auditado pela conformidade com ITAR ou outros controles de exportação e atualmente não oferece a capacidade de restringir traduções de acesso por país.

2.25.2 Controlos de criptografia dos EUA

O Weblate não contém nenhum código criptográfico, mas pode ser objeto de 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, irá incluir todas as funções criptográficas com exportação restrita:

- Em excesso de 56 bits para um algoritmo simétrico
- Fatorização de inteiros acima de 512 bits para um algoritmo assimétrico
- Cálculo de logaritmos discretos num grupo multiplicativo de um campo finito de tamanho maior do que 512 bits para um algoritmo assimétrico
- Logaritmos discretos num grupo diferente do que acima de 112 bits para um algoritmo assimétrico

O Weblate não tem nenhum recurso de ativação criptográfica, mas pode ser configurado de maneira sem ter nenhum código de criptografia envolvido. Os recursos criptográficos incluem:

- Acessar servidores remotos a usar protocolos seguros (HTTPS)
- Gerar assinaturas para commits de código (PGP)

Veja também:

Controles de Exportação (EAR) em Software de Código Aberto *(inglês)*

CAPÍTULO 3

Documentação de colaborador

3.1 Contribuir ao Weblate

Há dezenas de maneiras de melhorar o Weblate. Pode escolher uma com a qual se sente confortável, seja codificação, design gráfico, documentação, patrocínio ou uma ideia:

- *Relatando problemas no Weblate*
- *Comea çar a contribuir código para o Weblate*
- *Contributing to Weblate modules*
- *Traduzir o Weblate*
- *Contribua à documentação do Weblate*
- *Discussões do Weblate*
- *Financiar o desenvolvimento do Weblate*

3.1.1 Traduzir o Weblate

O Weblate está a ser continuamente traduzido a usar o próprio Weblate. Sinta-se à vontade para participar do esforço de disponibilizar o Weblate na maior quantidade possível de idiomas humanos. Isso traz o Weblate para mais perto dos utilizadores deles!

Se encontrar um possível erro na cadeia fonte, pode marcá-lo com um comentário no editor Weblate. Dessa forma, pode ser discutido e corrigido. Se tiver certeza, também pode clicar na ligação na secção *Localização do texto fonte* e enviar um PR com a sua correção.

3.1.2 Contribua à documentação do Weblate

É bem-vindo para melhorar a página de documentação de sua escolha. Faça isso facilmente a clicar no botão *Editar no GitHub* no canto superior direito da página.

Respeite essas diretrizes ao escrever:

1. Não remova parte da documentação se ela for válida.
2. Use uma linguagem clara e de fácil compreensão. Está a escrever documentos técnicos, não um poema. Nem todos os leitores de documentos são falantes nativos, fique atento.
3. Don't be afraid to ask if you are not certain. If you have to ask about some feature while editing, don't change its docs before you have the answer. This means: You change or ask. Don't do both at the same time.
4. Verifique as suas alterações a executar as ações descritas ao seguir os documentos.
5. Envie PR com alterações em pequenos pedaços para tornar mais fácil e rápido revisar e mesclar.
6. Se quiser reescrever e alterar a estrutura de um grande artigo, faça isso em duas etapas:
    1. Reescreva
    2. Depois que a reescrita for revisada, polida e mesclada, altere a estrutura dos parágrafos em outro PR.

Dica: Pode traduzir os documentos.

3.1.3 Estendendo definições embutidas de idioma

As definições de idioma estão no repositório weblate-language-data repository.

Está convidado a adicionar definições de idioma em falta a `languages.csv`, outros ficheiros são gerados a partir desse ficheiro.

3.1.4 Discussões do Weblate

Se tiver uma ideia e não tiver certeza se ela é adequada para um problema, não se preocupe. Pode entrar na comunidade de discussões do GitHub.

3.1.5 Financiar o desenvolvimento do Weblate

Pode impulsionar o desenvolvimento do Weblate na página de doação. Os fundos coletados lá são usados para permitir hospedagem de grátis para projetos de software livre e o desenvolvimento adicional do Weblate. Por favor, verifique a *página de doação* para obter opções, como metas de financiamento e as recompensas que obtém como um financiador orgulhoso.

Supporters who have funded Weblate

Lista de apoiadores do Weblate:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- Cassidy James
- Loic Dachary
- Marozed

- https://freedombox.org/
- GNU Solidario (GNU Health)
- BallotReady
- Richard Nespithal
- MyExpenses.Mobi

Gostaria de estar na lista? Veja as opções em Doar ao Weblate.

3.2 Comea çar a contribuir código para o Weblate

Entenda o código-fonte do Weblate a passar por *Código-fonte do Weblate*, *Frontend do Weblate* e *Componentes internos do Weblate*.

3.2.1 Comea çar com a base de código

Familiarize-se com a base de código do Weblate, a dar uma olhada nos bugs etiquetados como good first issue.

3.2.2 Execução local de Weblate

A abordagem mais confortável para começar a desenvolver Weblate é seguir …/admin/install/source. Isso lhe dará um virtualenv com fontes editáveis do Weblate.

1. Clone o código-fonte do Weblate:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Crie um virtualenv:

```
virtualenv .venv
.venv/bin/activate
```

3. Instale Weblate (para isso, vai precisar de algumas dependências do sistema, veja …/admin/install/source):

```
pip install -e .
```

3. Instale todas as dependências úteis para o desenvolvimento:

```
pip install -r requirements-dev.txt
```

4. Inicie um servidor de desenvolvimento:

```
weblate runserver
```

5. Dependendo da sua configuração, também quereria inciar workers do Celery:

```
./weblate/examples/celery start
```

6. Para executar um teste (veja *Teste local* para mais detalhes):

```
. scripts/test-database
./manage.py test
```

Veja também:

Instalando a partir do código-fonte

3.2.3 Executando Weblate localmente no Docker

Se tiver Docker e docker-compose instalados, pode pôr para funcionar o ambiente de desenvolvimento simplesmente a executar:

```
./rundev.sh
```

Isso irá criar uma imagem do Docker de desenvolvimento e iniciá-lo. Weblate está a ser executado em <http://127.0.0.1:8080/> e pode entrar como o utilizador `admin`, a usar `admin` como a palavra-passe. A nova instalação está vazia, então quererá seguir *Adicionando projetos e componentes de tradução*.

Os ficheiros `Dockerfile` e `docker-compose.yml` para isso estão localizados no diretório `dev-docker`.

O script também aceita alguns parâmetros; para executar testes, execute-o com o parâmetro `test` e então especifique qualquer parâmetro `test`, por exemplo executar apenas testes no módulo `weblate.machine`:

```
./rundev.sh test --failfast weblate.machine
```

Nota: Tenha cuidado para que os seus contentores Docker estejam prontos e em execução antes de fazer os testes. Pode verificar isso a executar o comando `docker ps`.

Para exibir os logs:

```
./rundev.sh logs
```

Para parar os contentores em segundo fundo, execute:

```
./rundev.sh stop
```

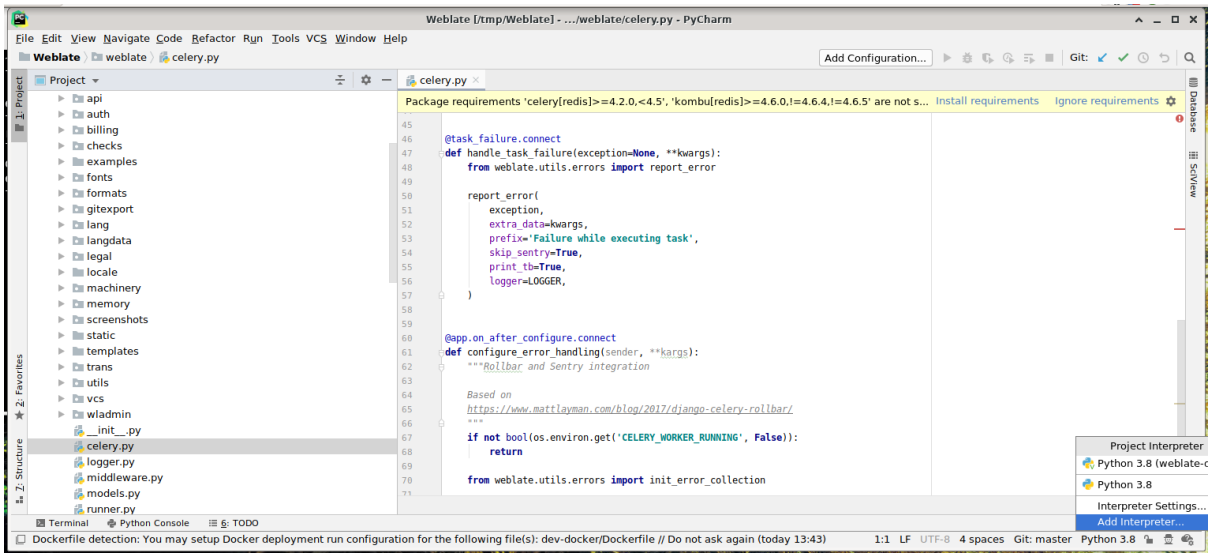
A execução do script sem argumentos vai recriar o contentor Docker e reiniciá-lo.

Nota: Esta não é uma configuração adequada para a produção, pois inclui vários hacks que são inseguros, mas que tornam o desenvolvimento mais fácil.

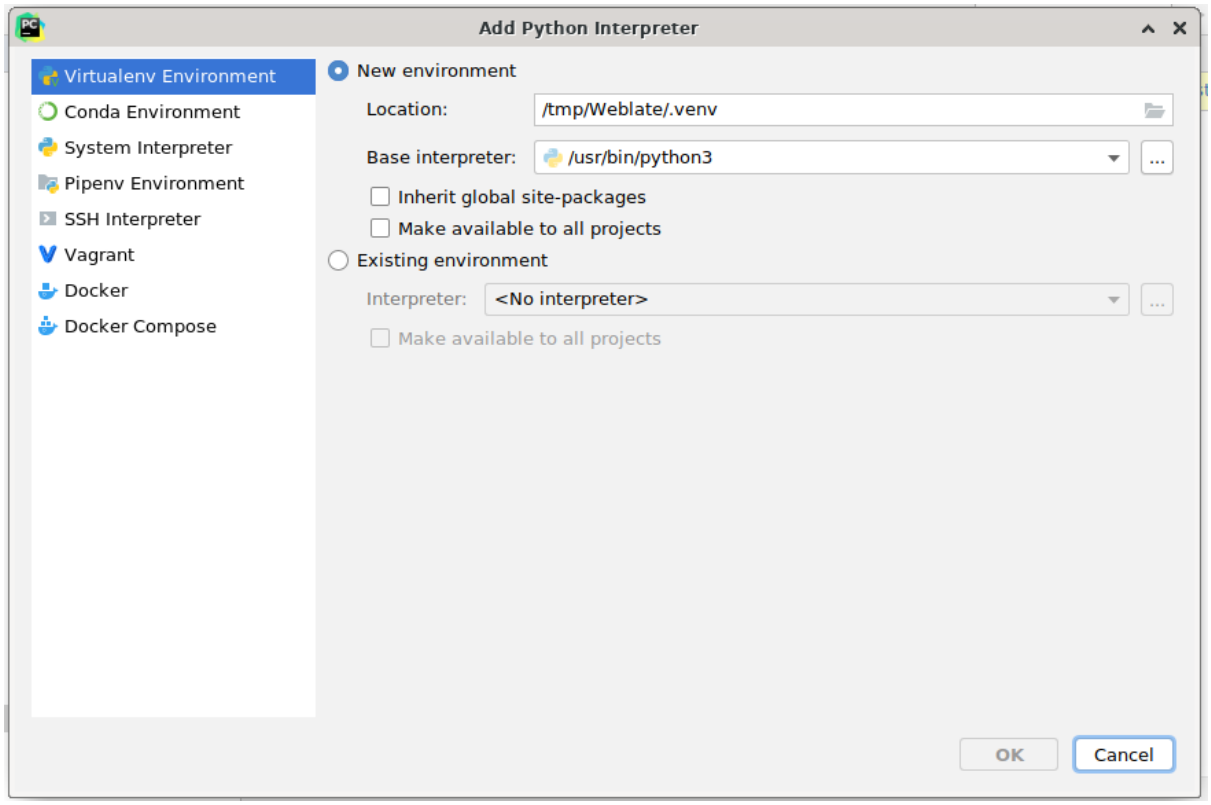
3.2.4 Codificando Weblate com PyCharm

PyCharm é uma IDE conhecida para Python, aqui estão algumas diretrizes para lhe ajudar a configurar o seu projeto Weblate nele.

Considerando que acabou de clonar o repositório GitHub, para uma pasta, basta abri-la com o PyCharm. Assim que a IDE estiver aberta, o primeiro passo é especificar o interpretador que deseja:

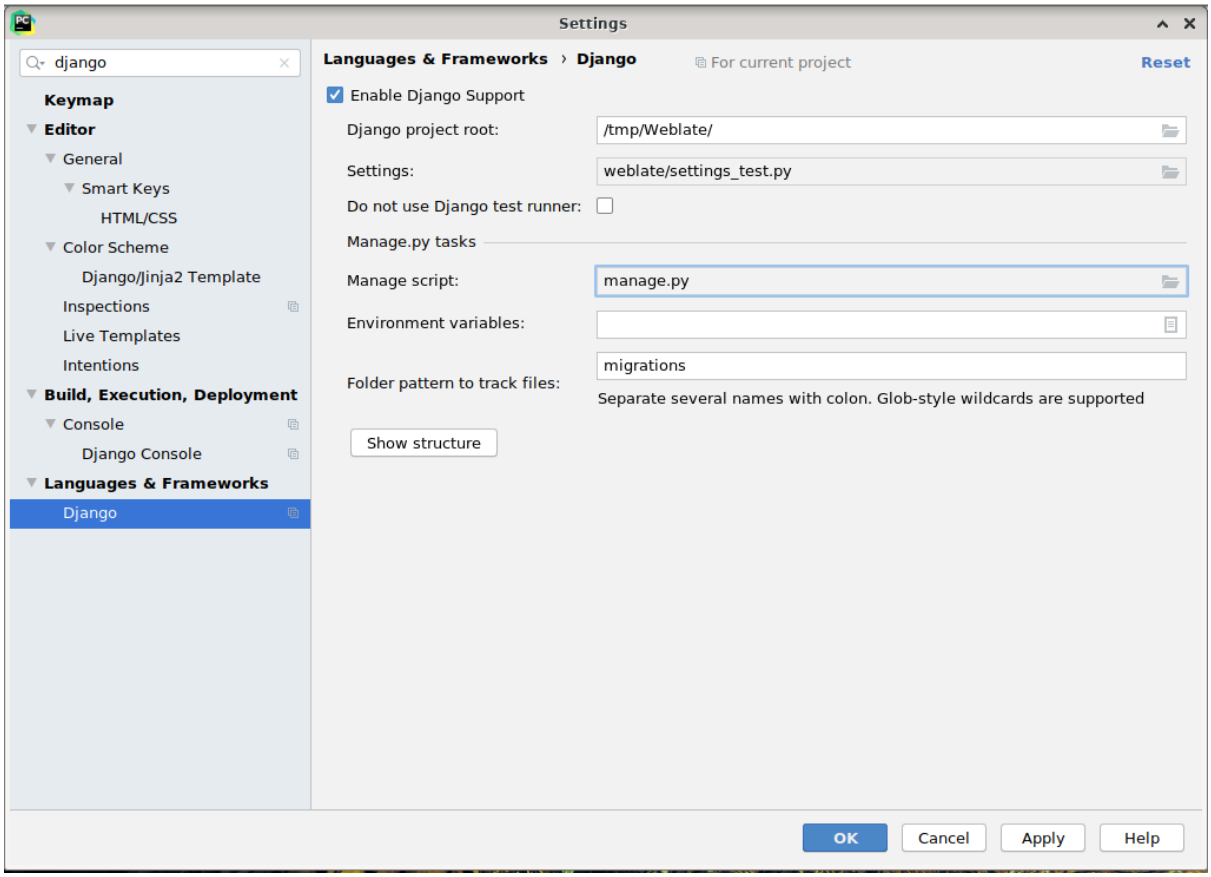


Pode optar por deixar PyCharm criar o virtualenv para si, ou selecionar um já existente:



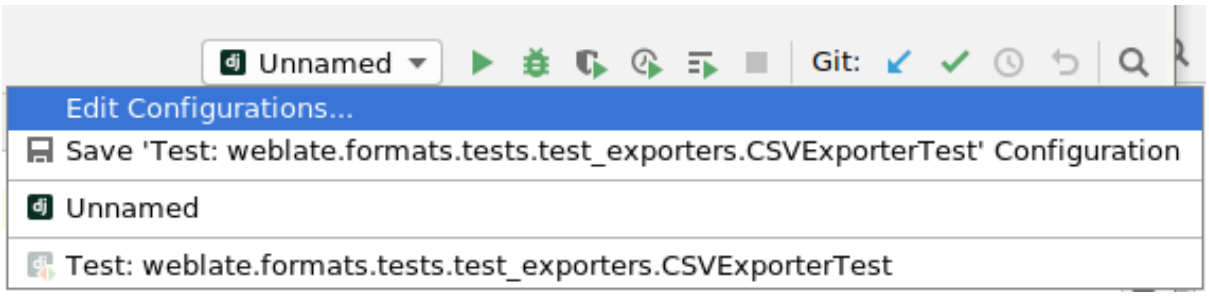
Não se esqueça de instalar as dependências quando o interpretador estiver configurado: faça isso através do console (o console do IDE vai usar diretamente o seu virtualenv por padrão, ou através da interface quando receber um aviso sobre dependências em falta.

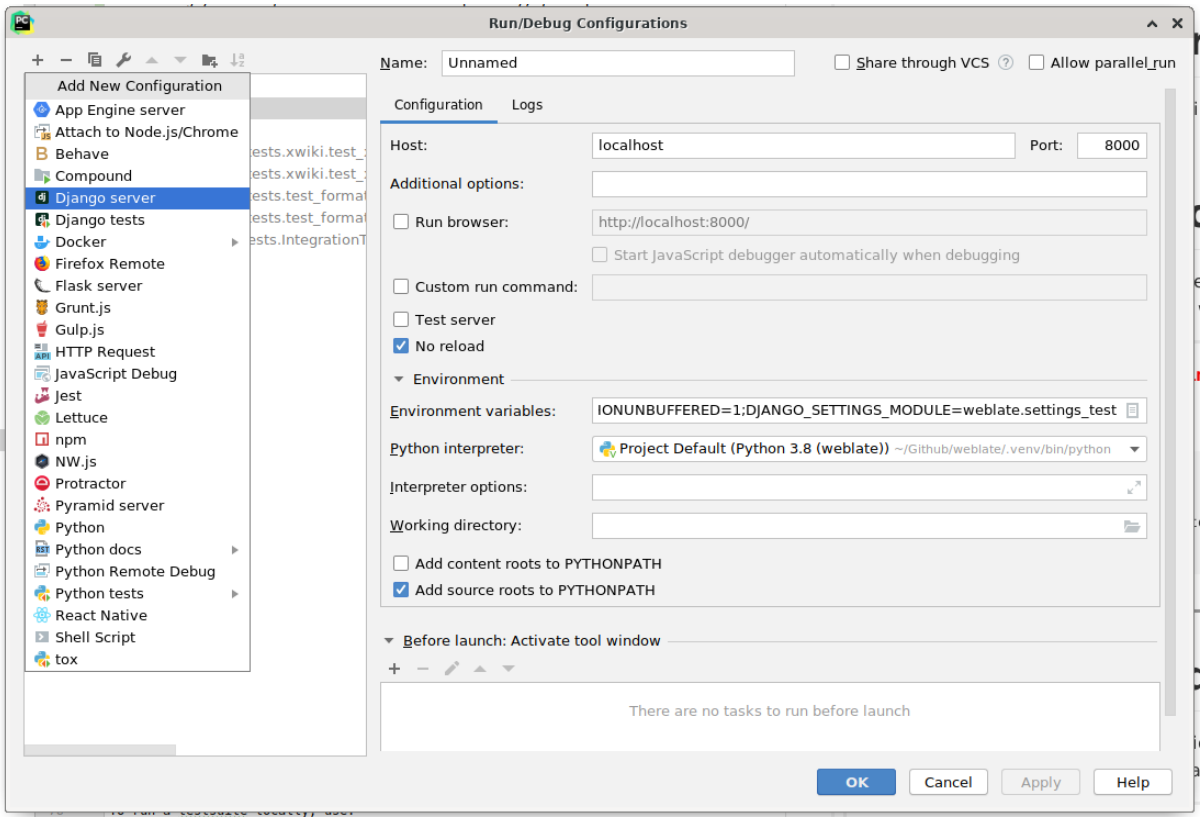
O segundo passo é definir a informação certa para usar o Django nativo dentro do PyCharm: a ideia é ser capaz de acionar imediatamente os testes unitários no IDE. Para isso, precisa especificar o caminho raiz do projeto Django e o caminho para as suas configurações:



Tenha cuidado, *Django project root* é a raiz real do repositório, não o subdiretório do Weblate. Sobre as configurações, poderia usar o `weblate/settings_test.py` do repositório, mas pode criar a sua própria configuração e configurá-la lá.

O último passo é executar o servidor e pôr pontos de interrupção no código para poder depurá-lo. Isto é feito a criar uma nova configuração do *Django Server*:





Dica: Tenha cuidado com a propriedade chamada *No reload*: ela evita do servidor ser recarregado automaticamente quando modificar os ficheiros. Isso permite que os pontos de interrupção de depuração existentes persistam, quando eles normalmente seriam descartados no recarregamento do servidor.

3.2.5 Inicializar a sua instância de desenvolvimento

Quererá usar `import_demo` para criar traduções de demonstração e `createadmin` para criar um utilizador administrador.

3.3 Código-fonte do Weblate

O Weblate é desenvolvido no GitHub. É bem-vindo para criar um fork do código e abrir pull requests. Patches em qualquer outra forma também são bem-vindos.

Veja também:

Confira *Componentes internos do Weblate* para ver como o Weblate se parece por dentro.

3.3.1 Coding guidelines

Qualquer código para Weblate deve ser escrito com Princípios de Segurança por Design (inglês) em mente.

Qualquer código deve vir com documentação explicando o comportamento. Não se esqueça de documentar métodos, blocos de código complexos ou recursos visíveis ao utilizador.

Qualquer novo código deve utilizar dicas de tipo **PEP 484**. Ainda não estamos verificando isso em nosso CI, pois o código existente ainda não os inclui.

3.3.2 Padrão de codificação e linting do código

O código deve seguir as diretrizes de codificação PEP-8 e deve ser formatado a usar o formatador de código **`black`**.

Para verificar a qualidade do código, pode usar o **:programa:`flake8`**, os plugins recomendados estão listados em `.pre-commit-config.yaml` e a configuração dele está em `setup.cfg`.

The easiest approach to enforce all this is to install pre-commit. The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the `requirements-lint.txt`) turn it on by running `pre-commit install` in Weblate checkout. This way all your changes will be automatically checked.

Também pode acionar a verificação manualmente, para verificar todos os ficheiros execute:

```
pre-commit run --all
```

3.4 Depuração do Weblate

Os bugs podem se comportar como falhas no aplicação ou como vários mau comportamentos. 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 ativar o modo de depuração, as exceções serão mostradas no navegador web. Isto é útil para problemas de depuração na interface web, mas não é adequado para o ambiente de produção porque tem consequências de desempenho e pode vazar dados privados.

Num ambiente de produção, use `ADMINS` para receber e-mails que conteem relatórios de erros ou configurar a coleta de erros a usar um serviço de terceiros.

Veja também:

Desativar o modo de depuração, *Configurar administradores corretamente*, *Coletando relatórios de erros*

3.4.2 Logs do Weblate

Weblate pode produzir registos (logs) detalhados do que está a acontecer em segundo fundo. Na configuração padrão, ele usa syslog e isso faz o log aparecer em `/var/log/messages` ou `/var/log/syslog` (a depender da configuração do seu daemon syslog).

O processo de Celery (veja *Tarefas de fundo a usar o Celery*) normalmente também produz os próprios registos. O exemplo de configurações de todo o sistema regista em vários ficheiros em `/var/log/celery/`.

Os contentores Docker registam a sua saída (como é habitual no mundo Docker), por isso pode olhar para os logs a utilizar `docker-compose logs`.

Veja também:

Amostra de configuração contém uma configuração de `LOGGING`.

3.4.3 Não a processar tarefas de segundo plano

Muitas coisas são feitas em segundo plano pelos workers do Celery. Se coisas como o envio de e-mails ou remoção de componentes não funcionarem, pode haver algum problema relacionado.

Coisas a verificar neste caso:

- Certifique-se que o processo do Celery está em execução, veja *Tarefas de fundo a usar o Celery*
- Verifique o status da fila do Celery em *Interface de gestão* ou a usar `celery_queues`
- Procure erros nos registos do Celery (veja *Logs do Weblate*)

3.4.4 Não a receber e-mails do Weblate

Pode verificar se o e-mail de saída está a funcionar corretamente a usar o comando de gestão `sendtestemail` (veja *Invocando comandos de gestão* para instruções sobre como invocá-lo em diferentes ambientes) ou a usar *Interface de gestão* sob a guia *Ferramentas*.

Estes enviam e-mails diretamente, para que isto verifique se a sua configuração SMTP está correta (veja *Configuração de e-mail de saída*). A maioria dos e-mails da Weblate são, no entanto, enviados em segundo plano e pode haver alguns problemas com o Celery envolvido também, veja *Não a processar tarefas de segundo plano* para depuração disso.

3.4.5 Análise de falhas de aplicação

Caso a aplicação falhe, é útil recolher o máximo de informação possível sobre a falha. Isso pode ser feito a usar serviços de terceiros que podem coletar tais informações automaticamente. Pode encontrar informações sobre como configurar isso em *Coletando relatórios de erros*.

3.4.6 Falhas silenciosas

Muitas tarefas são descarregadas para o Celery para processamento em segundo plano. As falhas não são exibidas na interface do utilizador, mas aparecem nos logs do Celery. A configuração em *Coletando relatórios de erros* ajuda a perceber essas falhas mais facilmente.

3.4.7 Problemas de desempenho

Caso o Weblate tenha um mau desempenho em algum cenário, por favor, recolha os registos relevantes a mostrar o problema e qualquer coisa que possa ajudar a descobrir onde o código pode ser melhorado.

Caso alguns pedidos demorem muito tempo sem nenhuma indicação, pode querer instalar dogslow juntamente com *Coletando relatórios de erros* e obter tracebacks detalhados na ferramenta de coleta de erros.

Caso o desempenho lento esteja vinculado ao banco de dados, também pode ativar o log de todas as consultas do banco de dados usando a seguinte configuração após ativar `DEBUG`:

```
LOGGING["loggers"]["django.db.backends"] = {"handlers": ["console"], "level":
↪"DEBUG"}
```

3.5 Componentes internos do Weblate

Nota: Este capítulo lhe dará uma visão geral básica dos componentes internos do Weblate.

Weblate deriva a maior parte da estrutura de código dele e é baseado no Django.

3.5.1 Estrutura de diretórios

Visão geral rápida da estrutura de diretórios do repositório principal Weblate:

`docs`
Código-fonte desta documentação, que pode ser construída a usar Sphinx.

`dev-docker`
Código do Docker para executar o servidor de desenvolvimento, veja *Executando Weblate localmente no Docker*.

`weblate`
Código-fonte do Weblate como uma aplicação Django, veja *Componentes internos do Weblate*.

`weblate/static`
Ficheiro de cliente (CSS, Javascript e imagens), veja *Frontend do Weblate*.

3.5.2 Módulos

O Weblate consiste em vários aplicações Django (alguns opcionais, veja *Módulos opcionais do Weblate*):

`accounts`

Conta de utilizador, 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 *Faturação*.

`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 ficheiro baseada no translate-toolkit.

`gitexport`

O módulo opcional *Exportador git*.

`lang`

Módulo que define idioma e modelos plurais.

`legal`

O módulo opcional *Legal*.

`machinery`

Integração de serviços de tradução de máquina.

`memory`

Memória de tradução embutida, veja *Memória de Tradução*.

`screenshots`

Gestão de capturas de ecrã e módulo OCR.

`trans`

Módulo principal que trata das traduções.

`utils`

Várias utilidades de ajuda.

`vcs`

Abstração do sistema de controle de versão.

`wladmin`

Personalização da interface administrativa do Django.

3.6 Desenvolver extensões

Extensões são uma forma de personalizar o fluxo de trabalho de localização no Weblate.

class `weblate.addons.base.`**`BaseAddon`**(*storage=None*)

Classe base para as extensões do Weblate.

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 feito um commit para o repositório.

post_push(*component*)

Gancho acionado após repositório ter o push feito para o upstream.

post_update(*component*, *previous_head: str*, *skip_push: bool*)

Gancho acionado após o repositório ser atualizado a partir do upstream.

Parâmetros

- **previous_head**(*str*) – HEAD do repositório antes da atualização, pode estar vazio na clonagem inicial.
- **skip_push** (*bool*) – 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 feito um commit para o repositório.

pre_push(*component*)

Gancho acionado antes do repositório ter feito push para o upstream.

pre_update(*component*)

Hook acionado antes do repositório ser atualizado a partir do upstream.

save_state()

Salva informações do estado da extensão.

store_post_load(*translation*, *store*)

Gancho acionado após um ficheiro ser analisado.

Ele recebe uma instância de uma classe de formato de ficheiro como um argumento.

Isso é útil para modificar os parâmetros de classe de formato de ficheiro, por exemplo, ajustar como o ficheiro será salvo.

unit_pre_create(*unit*)

Gancho acionado antes que a nova unidade seja criada.

Aqui está um exemplo de extensão:

```
#
# Copyright © 2012-2022 Michal Čihař <michal@cihar.com>
#
# This file is part of Weblate <https://weblate.org/>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#


from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT


class ExampleAddon(BaseAddon):
```

(continues on next page)

(continuação da página anterior)

```
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {"file_format": {"po", "po-mono"}}
    # List of events add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
    # Detailed add-on description
    description = _("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Frontend do Weblate

O frontend atualmente é construído a usar Bootstrap, jQuery e algumas bibliotecas de terceiros.

3.7.1 Navegadores suportados

Weblate oferece suporte às versões mais recentes e estáveis de todos os principais navegadores e plataformas.

Navegadores alternativos que usam a versão mais recente do WebKit, Blink ou Gecko, seja diretamente ou por meio da API de visualização da web da plataforma, não são explicitamente suportados. No entanto, o Weblate deve (na maioria dos casos) ser exibido e funcionar corretamente nesses navegadores também.

Navegadores mais antigos podem funcionar, mas alguns recursos podem ser limitados.

3.7.2 Gestão de dependências

O gestor de pacotes yarn é usado para atualizar bibliotecas de terceiros. A configuração reside em `scripts/yarn` e há um script wrapper `scripts/yarn-update` para atualizar as bibliotecas, construí-las e copiá-las para os locais corretos em `weblate/static/vendor`, onde todo o terceiro código parcialmente frontend está localizado. O código específico do Weblate deve ser posto diretamente em `weblate/static` ou em diretórios específico do recurso (por exemplo, `weblate/static/editor`).

Adicionar uma nova biblioteca de terceiros geralmente consiste em:

```
# Add a yarn package
yarn --cwd scripts/yarn add PACKAGE
# Edit the script to copy package to the static folder
edit scripts/yarn-update
# Run the update script
./scripts/yarn-update
# Add files to git
git add .
```

3.7.3 Estilo de codificação

Weblate depende do Prettier para a formatação do código para ficheiros JavaScript e CSS.

Também usamos ESLint para verificar o código JavaScript.

3.7.4 Tradução

Se precisar de qualquer texto visível para o utilizador no código do frontend, ele deve ser localizável. Na maioria dos casos, tudo que precisa é envolver o seu texto dentro da função `gettext`, mas existem recursos mais complexos disponíveis:

```
document.write(gettext('this is to be translated'));

var object_count = 1 // or 0, or 2, or 3, ...
s = ngettext('literal for the singular case',
        'literal for the plural case', object_count);

fmts = ngettext('There is %s object. Remaining: %s',
        'There are %s objects. Remaining: %s', 11);
s = interpolate(fmts, [11, 20]);
// s is 'There are 11 objects. Remaining: 20'
```

Veja também:

Tópico sobre tradução na documentação do Django

3.7.5 Ícones

Weblate atualmente usa ícones de design de material. Caso esteja a procurar por um novo símbolo, verifique Material Design Icons ou Material Design Resources.

Além disso, existe `scripts/optimize-svg` para reduzir o tamanho do SVG, já que a maioria dos ícones são embutidos no HTML para permitir estilização dos caminhos.

3.8 Relatando problemas no Weblate

O rastreador de problemas do Weblate está hospedado no GitHub.

Sinta-se à vontade para relatar quaisquer problemas que tenha ou sugerir melhorias para o Weblate lá. Existem vários modelos preparados para lhe orientar confortavelmente durante o relatório de problemas.

Se o que encontrou é um problema de segurança no Weblate, consulte a secção *Problemas de segurança* abaixo.

Se não tem certeza sobre o seu relatório de bug ou solicitação de recurso, pode tentar *Discussões do Weblate*.

3.8.1 Problemas de segurança

A fim de dar à comunidade tempo para responder e fazer upgrade, recomendamos enfaticamente que relate todos os problemas de segurança em particular. O HackerOne é usado para lidar com problemas de segurança e pode ser relatado diretamente em HackerOne. Depois de enviá-lo lá, a comunidade terá tempo limitado, mas suficiente para resolver o incidente.

Como alternativa, relate para security@weblate.org, que termina no HackerOne também.

Se não quiser usar o HackerOne, por qualquer motivo, pode enviar o relatório por e-mail para michal@cihar.com. Pode optar por criptografá-lo a usar esta chave PGP *3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D*. Também pode obter a chave PGP em Keybase.

Nota: O Weblate depende de componentes de terceiros para muitas coisas. Caso encontre uma vulnerabilidade a afetar um desses componentes em geral, relate-o diretamente ao respectivo projeto.

Alguns deles são:

- Django
- framework Django REST
- Python Social Auth

3.9 Conjunto de testes e integração contínua do Weblate

Existem conjuntos de teste para a maior parte do código atual, aumente a cobertura a adicionar casos de teste para qualquer nova funcionalidade e verifique se funciona.

3.9.1 Integração contínua

Os resultados do teste atual encontram-se em GitHub Actions e a cobertura é relatada em Codecov.

Existem vários trabalhos para verificar diferentes aspectos:

- Testes de unidade
- Construção de documentação e ligações externas
- Testes de migração de todas as versões com suporte
- Linting de código
- Verificação de configuração (garante que os ficheiros dist gerados não percam nada e possam ser testados)

A configuração do CI está no diretório `.github/workflows`. Ele usa muito scripts auxiliares armazenados no diretório `ci`. Os scripts também podem ser executados manualmente, mas eles requerem várias variáveis de ambiente, principalmente a definir o ficheiro de configurações do Django para usar e conexão com o banco de dados. A definição de exemplo disso está em `scripts/test-database`:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

A execução simples pode ser semelhante a:

```
. scripts/test-database
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Teste local

Para executar um conjunto de testes localmente, use:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Dica: 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 que as suas configurações estarem configuradas para usar `weblate`, os testes usarão o banco de dados `test_weblate`. Veja *Configuração de banco de dados para o Weblate* para instruções de configuração.

O `weblate/settings_test.py` é usado no ambiente CI também (veja *Integração contínua*) e pode ser ajustado a usar variáveis de ambiente:

```
# Simple way to configure test database from environment

# Database backend to use postgresql / mysql / mariadb
export CI_DATABASE=${1:-postgresql}

# Database server configuration
export CI_DB_USER=weblate
export CI_DB_PASSWORD=weblate
export CI_DB_HOST=127.0.0.1

# Django settings module to use
export DJANGO_SETTINGS_MODULE=weblate.settings_test
```

Antes de executar os testes, deve coletar ficheiros estáticos, pois alguns testes contam com a presença deles:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

Também pode especificar testes individuais para executar:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Dica: Os testes também podem ser executados dentro de um contentor Docker do programador, consulte *Executando Weblate localmente no Docker*.

Veja também:

Veja :doc:`django:topics/testing/index`para mais informações sobre a execução e escrita de testes para Django.

3.10 Esquemas de dados

Weblate usa JSON Schema para definir o layout de ficheiros JSON externos.

3.10.1 Esquema de memória de tradução Weblate

https://weblate.org/schemas/weblate-memory.schema.json			
tipo	*array*		
itens	*O item de memória de tradução*		
	tipo	*objeto*	
	propriedades		
	• **category**	*A categoria de cadeia*	
		1 é global, 2 é compartilhado, 10000000+ são específicos de projeto, 20000000+ são específicos do utilizador	
		tipo	*inteiro*
		exemplos	1
		minímo	0
		predefinido	1
	• **origem**	*A origem do texto*	
		Nome do ficheiro ou componente	
		tipo	*cadeia*
		exemplos	test.tmx
			projeto/componente
		predefinido	
	• **source**	*A cadeia fonte*	
		tipo	*cadeia*
		exemplos	Olá
		minLength	1
		predefinido	
	• **idioma_ fonte**	**O idioma fonte**	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		tipo	*cadeia*
		exemplos	en
		pattern	^[^]+$
		predefinido	
	• **target**	*A cadeia de destino*	
		tipo	*cadeia*
		exemplos	Ahoj
		minLength	1
		predefinido	
	• **idioma_de_destino**	*O idioma de destino*	
		ISO 639-1 / ISO 639-2 / IETF BCP 47	
		tipo	*cadeia*
		exemplos	cs
		pattern	^[^]+$
		predefinido	
	additionalProperties	False	
definições			

Veja também:

Memória de Tradução, `dump_memory`, `import_memory`

3.10.2 Exportação de dados de utilizadores do Weblate

https://weblate.org/schemas/weblate-userdata.schema.json			
tipo	*objeto*		
propriedades			
• **basic**	*Básico*		
	tipo	*objeto*	
	propriedades		
	• **nome do utilizador**	*Nome do utilizador*	
		tipo	*cadeia*
		exemplos	administrador
		predefinido	
	• **nome_completo**	*Nome completo*	
		tipo	*cadeia*
		exemplos	Admin Weblate
		predefinido	
	• **email**	*E-mail*	
		tipo	*cadeia*
		exemplos	noreply@example.com
		predefinido	
		format	e-mail
	• **data_de_adesão**	*Data de adesão*	
		tipo	*cadeia*
		exemplos	2019-11-18T18:53:54.862Z
		predefinido	
		format	date-time
	additionalProperties	False	
• **profile**	*Perfil*		
	tipo	*objeto*	
	propriedades		
	• **idioma**	*Idioma*	
		tipo	*cadeia*
		exemplos	cs
		pattern	^[^]*$
		predefinido	
	• **sugerido**	*Quantidade de cadeias fonte*	
		tipo	*inteiro*
		exemplos	1
		predefinido	0
	• **traduzido**	*Quantidade de cadeias traduzidas*	
		tipo	*inteiro*
		exemplos	24
		predefinido	0
	• **enviado**	*Quantidade de capturas de ecrã enviadas*	
		tipo	*inteiro*
		exemplos	1
		predefinido	0
	• **hide_completed**	*Ocultar as traduções completas no painel*	
		tipo	*booleano*
		exemplos	False
		predefinido	True
	• **secondary_in_zen**	*Mostrar traduções secundárias no modo Zen*	
		tipo	*booleano*
		exemplos	True
		predefinido	True

continues on next page

Table 2 – continuação da página anterior

	• **hide_source_secondary**	*Ocultar a fonte se existir uma tradução secundária*		
		tipo	*booleano*	
		exemplos	False	
		predefinido	True	
	• **hiperligação_do_editor**	*Hiperligação_do_editor*		
		tipo	*cadeia*	
		exemplos		
		pattern	^.*$	
		predefinido		
	• **modo_de_tradução**	*Modo do editor de tradução*		
		tipo	*inteiro*	
		exemplos	0	
		predefinido	0	
	• **zen_mode**	*Modo de editor Zen*		
		tipo	*inteiro*	
		exemplos	0	
		predefinido	0	
	• **carateres_especiais**	*Carateres especiais*		
		tipo	*cadeia*	
		exemplos		
		pattern	^.*$	
		predefinido		
	• **vista_do_painel**	*Vista predefinida do painel*		
		tipo	*inteiro*	
		exemplos	1	
		predefinido	0	
	• **dashboard_component_list**	*Lista predefinida de componentes*		
		predefinido	null	
		anyOf	tipo	*null*
			tipo	*inteiro*
	• **languages**	*Idiomas traduzidos*		
		tipo	*array*	
		predefinido		
		itens	*Código do idioma*	
			tipo	*cadeia*
			exemplos	cs
			pattern	^.*$
			predefinido	
	• **idiomas_secundários**	*Idiomas secundários*		
		tipo	*array*	
		predefinido		
		itens	*Código do idioma*	
			tipo	*cadeia*
			exemplos	sk
			pattern	^.*$
			predefinido	
	• **watched**	*Projetos vigiados*		
		tipo	*array*	
		predefinido		
		itens	*Slug do projeto*	
			tipo	*cadeia*
			exemplos	weblate
			pattern	^.*$
			predefinido	
	additionalProperties	False		

continues on next page

Table 2 – continuação da página anterior

• **auditlog**	*Registo de auditoria*			
	tipo	*array*		
	predefinido			
	itens	*Itens*		
		tipo	*objeto*	
		propriedades		
		• **address**	*Endereço IP*	
			tipo	*cadeia*
			exemplos	127.0.0.1
			pattern	^.*$
			predefinido	
		• **agente_do_utilizador**	*Agente do utilizador*	
			tipo	*cadeia*
			exemplos	PC / Linux / Firefox 70.0
			pattern	^.*$
			predefinido	
		• **timestamp**	*Carimbo de data e hora*	
			tipo	*cadeia*
			exemplos	2019-11-18T18:58:30.845Z
			predefinido	
			format	date-time
		• **atividade**	*Atividade*	
			tipo	*cadeia*
			exemplos	sessão
			pattern	^.*$
			predefinido	
		additionalProperties	False	
definições				

Veja também:

Perfil do utilizador, `dumpuserdata`

3.11 Criando lançamento do Weblate

3.11.1 Agenda de lançamentos

O Weblate tem um ciclo de lançamento de dois meses para lançamentos (x.y). Geralmente, eles são seguidos por várias versões de correções de bugs para corrigir problemas que ocorrem neles (x.y.z).

A mudança na versão principal indica que o processo de atualização não pode ignorar esta versão - sempre deve atualizar para x.0 antes de atualizar para versões x.y superiores.

Veja também:

Atualizando o Weblate

3.11.2 Planejamento de lançamento

Os recursos para os próximos lançamentos são coletados a usar marcos do GitHub, pode ver nosso roteiro em <https://github.com/WeblateOrg/weblate/milestones>.

3.11.3 Processo de lançamento

Coisas para verificar antes de fazer um lançamento:

1. Verificar novos idiomas traduzidos a usar **`./scripts/list-translated-languages`**.
2. Definir a versão final a usar **`./scripts/prepare-release`**.
3. Make sure screenshots are up to date **`make -j 12 -C docs update-screenshots`**.
4. Junte todas as traduções pendentes a usar **`wlc push; git remote update; git merge origin/weblate`**

Realizar o lançamento:

5. Criar um lançamento a usar **`./scripts/create-release --tag`** (veja abaixo os requisitos).

Etapas manuais pós-lançamento:

6. Atualizar a imagem Docker.
7. Fechar o marco do GitHub.
8. Assim que a imagem Docker for testada, adicionar uma tag e fazer o push dela.
9. Atualizar um chart Helm para a nova versão.
10. Incluir a nova versão em `.github/workflows/migrations.yml` para cobri-la nos testes de migração.
11. Incrementar a versão nas ligações de descarregas do site.
12. Incrementar a versão no repositório a usar **`./scripts/set-version`**.

Para criar tags a usar o script **`./scripts/create-release`**, precisará do seguinte:

- GnuPG com chave privada usada para assinar o lançamento
- Acesso por push a repositórios git Weblate (ele envia tags)
- Ferramenta **`hub`** configurada e acesso para criar lançamentos no repositório Weblate
- Acesso SSH ao servidor de descarregas Weblate (as descarregas do site são copiados lá)

3.12 Segurança e privacidade

Dica: No Weblate, a segurança mantém um ambiente que valoriza a privacidade dos nossos utilizadores.

O desenvolvimento do Weblate segue as Melhores Práticas da Core Infrastructure Initiative da Linux Foundation.

Veja também:

Problemas de segurança

3.12.1 Rastreamento de dependências para vulnerabilidades

Problemas de segurança em nossas dependências são monitorados a usar Dependabot. Isso cobre as bibliotecas Python e JavaScript e a versão estável mais recente tem as dependências delas 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 contentor Docker

Os contentores Docker são verificados regularmente a usar Anchore e Trivy.

Isso nos permite detetar vulnerabilidades antecipadamente e lançar melhorias rapidamente.

Pode obter os resultados dessas varreduras no GitHub – eles são armazenados como artefatos em nosso CI no formato SARIF (Static Analysis Results Interchange Format).

Veja também:

Integração contínua

3.13 Contributing to Weblate modules

Além do repositório principal, o Weblate consiste em vários módulos Python. Todos estes seguem a mesma estrutura e esta documentação abrange todos eles.

For example, this covers:

- wlc, biblioteca Python cliente, veja *Cliente Weblate*
- translation-finder, usado para descobrir ficheiros traduzíveis no repositório
- language-data, language definitions for Weblate, see *Definições de idioma*

3.13.1 Coding guidelines

Qualquer código para Weblate deve ser escrito com Princípios de Segurança por Design (inglês) em mente.

Qualquer código deve vir com documentação explicando o comportamento. Não se esqueça de documentar métodos, blocos de código complexos ou recursos visíveis ao utilizador.

Qualquer novo código deve utilizar dicas de tipo **PEP 484**. Ainda não estamos verificando isso em nosso CI, pois o código existente ainda não os inclui.

3.13.2 Running tests

Os testes são executados usando `py.test`. Primeiro precisa instalar os requisitos de teste:

```
pip install -r requirements-test.txt
```

Pode então executar o testsuite no checkout do repositório:

```
py.test
```

Veja também:

A integração do CI é muito semelhante a *Conjunto de testes e integração contínua do Weblate*.

3.13.3 Padrão de codificação e linting do código

O código deve seguir as diretrizes de codificação PEP-8 e deve ser formatado a usar o formatador de código **`black`**.

Para verificar a qualidade do código, pode usar o **:programa:`flake8`**, os plugins recomendados estão listados em `.pre-commit-config.yaml` e a configuração dele está em `setup.cfg`.

The easiest approach to enforce all this is to install pre-commit. The repository contains configuration for it to verify the committed files are sane. After installing it (it is already included in the `requirements-lint.txt`) turn it on by running `pre-commit install` in Weblate checkout. This way all your changes will be automatically checked.

Também pode acionar a verificação manualmente, para verificar todos os ficheiros execute:

```
pre-commit run --all
```

Veja também:

Código-fonte do Weblate

3.14 Sobre o Weblate

3.14.1 Objetivos do projeto

Web-based continuous localization tool with tight *Integração de controlo de versões* supporting a wide range of *file formats*, making it easy for translators to contribute.

3.14.2 Nome do projeto

«Weblate» é uma palavra-valise das palavras «web» e «translate».

3.14.3 Site da Web do Projeto

A página inicial é https://weblate.org e há um serviço hospedado na nuvem em https://hosted.weblate.org. A documentação pode ser lida em https://docs.weblate.org.

3.14.4 Logotipos do projeto

Os logotipos do projeto e outros gráficos estão disponíveis em https://github.com/WeblateOrg/graphics.

3.14.5 Liderança

Este projeto é mantido por Michal Čihař, que pode ser alcançado em michal@cihar.com.

3.14.6 Autores

Weblate foi iniciado por Michal Čihař. Desde a criação dele, em 2012, milhares de pessoas contribuíram.

3.15 Licença

Copyright © 2012–2022 Michal Čihař michal@cihar.com

Este programa é um software livre: pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU, conforme publicado pela Free Software Foundation, seja a versão 3 da Licença, ou (ao seu critério) qualquer versão posterior.

Este programa é distribuído na esperança de que ele seja útil, mas sem qualquer garantia; sem sequer a garantia implícita de COMERCIALIZAÇÃO ou ADEQUAÇÃO PARA UM PROPÓSITO ESPECÍFICO. Consulte a Licença Pública Geral GNU para obter mais detalhes.

Deve ter recebido uma cópia da Licença Pública Geral GNU junto com este programa. Caso contrário, veja <https://www.gnu.org/licenses/>.

CAPÍTULO 4

Histórico de alterações

4.1 Weblate 4.14.1

Released on September 15th 2022.

- Fixed generating project backups in some situations.
- Improved error reporting on file upload.
- Fetch all user verified e-mails from GitHub during authentication.
- Avoid matching glossary terms on context or keys.
- Added notifications for string removals.
- Improved management of untranslatable terms in glossary.
- List number of team members on team management page.
- Add group management interface.
- Always show review stats when reviews are enabled.
- Added searching support in units API.
- Fixed progress bar display for read-only strings in the review workflow.
- Improved Burmese punctuation check.
- Fixed garbage collecting of metrics data.

All changes in detail.

4.2 Weblate 4.14

Released on August 22nd 2022.

- Track add-on changes in a history.
- Fixed parsing translation from Windows RC, HTML and text files.
- Extended language code style configuration options.
- Added support for plurals updated in the recent CLDR releases.
- Reduced memory usage while updating components with a lot of translations.
- Added support for translation domain in SAP Translation Hub.
- Allow absolute links in source string locations.
- Improved operation behind some reverse proxies.
- Extended API to cover translation memory.
- Improved document translation workflow.
- Improved reliability of HTML and text files translation.
- Added support for project level backups.
- Improved performance and memory usage of translation memory lookups.

All changes in detail.

4.3 Weblate 4.13.1

Released on July 1st 2022.

- Fixed tracking suggestions in history.
- Fixed parsing reverse proxy info from Cloudflare.
- Make parse error lock a component from translating.
- Fixed configuring intermediate file in the discovery add-on.
- Fixed DeepL translations behavior with placeholders.
- Fixed untranslating strings via API.
- Added support for removing user from a group via API.
- Fixed audit log for user invitation e-mails.
- Fixed flag names for Java formatting strings.

All changes in detail.

4.4 Weblate 4.13

Released on June 15th 2022.

- Changed behavior of updating language names.
- Added pagination to projects listing.
- A API para criar novas unidades agora retorna informações sobre a unidade recém-criada.
- Component discovery now supports configuring an intermediate language.

- Added fixed encoding variants to CSV formats.
- Changed handling of context and location for some formats to better fit underlying implementation.
- Added support for ResourceDictionary format.
- Improved progress bar colors for color blind.
- Fixed variants cleanup on string removal.
- Compatibility with Django 4.1.
- Added support for storing escaped XML elements in XLIFF.
- Improved formatting of placeholder check errors.
- Redirect /.well-known/change-password to /accounts/password/.
- Machine translation services are now configurable per project.
- Added separate permission for resolving comments and grant it to the *Review strings* role.
- Added support for storing alternative translations in the CSV file.
- The placeholders check can now be case-insensitive as well.

All changes in detail.

4.5 Weblate 4.12.2

Released on May 11th 2022.

- Fixed rebuilding project translation memory for some components.
- Fixed sorting components by untranslated strings.
- Fixed possible loss of translations while adding new language.
- Garante que a chave SSH do Weblate seja gerada durante as migrações.

All changes in detail.

4.6 Weblate 4.12.1

Released on April 29th 2022.

- Fixed pull request message title.
- Improved syntax error handling in Fluent format.
- Fixed avatar display in notification e-mails.
- Add support for web monetization.
- Fixed removal of stale source strings when removing translations.

All changes in detail.

4.7 Weblate 4.12

Released on April 20th 2022.

- Added support for Amharic in *Ponto final não correspondente*.
- Added support for Burmese in *Ponto de interrogação não correspondente*.
- Extended options of the *Geração da pseudolocalidade* add-on.
- Adicionado o marcador `ignore-all-checks` para ignorar todas as verificações de qualidade num texto.
- Avoid *Geração da pseudolocalidade* add-on to trigger failing checks.
- Added support for *Gitea pull requests*.
- Added Linux style language code to *Estilo de código de idioma*.
- Added support for rebuilding project translation memory.
- Improved API for creating components from a file.
- Add copy and clone buttons to other translations.
- Make merge request message configurable at component level.
- Improved maximal length restriction behavior with XML tags.
- Fixed loading Fluent files with additional comments.

All changes in detail.

4.8 Weblate 4.11.2

Released on March 4th 2022.

- Fixed corrupted MO files in the binary release.

All changes in detail.

4.9 Weblate 4.11.1

Released on March 4th 2022.

- Corrigida a falta de sanitização de argumentos para Git e Mercurial – CVE-2022-23915, consulte `GHSA-3872-f48p-pxqj <https://github.com/WeblateOrg/weblate/security/advisories/GHSA-3872-f48p-pxqj> `_ para mais detalhes.
- Fixed loading fuzzy strings from CSV files.
- Added support for creating teams using the API.
- Fixed user mention suggestions display.
- The project tokens access can now be customized.

All changes in detail.

4.10 Weblate 4.11

Released on February 25th 2022.

- Corrige XSS armazenado - CVE-2022-24710, veja GHSA-6jp6-9rf9-gc66 para mais detalhes.
- Corrigida a instalação de extensão usando API.
- Renomeada *Textos que necessitam ação* para *Textos não finalizados*.
- Corrigidos falso-positivos de *Sintaxe de MessageFormat do ICU*.
- Indicate lock and contributor agreement on other occurrences listing.
- Corrigida a atualização de ficheiros PO com textos obsoletos ou plurais ausentes.
- Aprimorada a compatibilidade da extensão de squash com Gerrit.
- Inicializar automaticamente os idiomas do utilizador à base do cabeçalho :http:header:"Accept-Language".
- Improved error handling on string removal.
- Weblate now requires Python 3.7 or newer.
- Fixed some write operations with project token authentication.
- Fixed string state tracking when the strings changes in the repository.
- Track string changes from the repository.
- Cabeçalho fixo na listagem de traduções para melhorar a navegação.
- Fixed untranslating strings in *Propriedades Java*.
- Corrigida a operação do Git com nomes de ramos não ASCII.
- New add-on *Preencher a tradução com fonte*.
- Added *Merge without fast-forward* *Estilo de união*.
- Fixed *Tradução automática* add-on trigger on newly added strings.
- Improved punctuation checks for Burmese.
- Adicionado suporte para definir equipas personalizadas a nível de projeto para conceder acesso a utilizadores, veja *A gerir controle de acesso por projeto*.
- Added documentation links to alerts.
- O contentor Docker ativa automaticamente TLS/SSL para e-mail de saída quando necessário.
- Added support for searching for resolved comments.
- Added support for borgbackup 1.2.
- Fixed applying of *Automatically translated* label.

All changes in detail.

4.11 Weblate 4.10.1

Lançado a 22 de dezembro de 2021.

- Alterações documentadas introduzidas pela atualização ao Django 4.0.
- Corrigida a exibição do rótulo *Traduzido automaticamente*.
- Corrigida a exibição de API do ramo em componentes com um repositório compartilhado.
- Melhorada a análise no alerta de push com falha.
- Corrigida a edição manual da página ao navegar pelas alterações.

- Merolhada a precisão da *Letra Kashida utilizada*.
- O contentor Docker do Weblate agora usa Python 3.10.

All changes in detail.

4.12 Weblate 4.10

Lançado a 16 de dezembro de 2021.

- Adicionado suporte para formalidade e espaços reservados com DeepL.
- A edição em massa e a pesquisa e substituição agora estão disponíveis no nível do projeto e do idioma.
- Adicionada filtragem para pesquisa e substituição.
- Corrigido: o privilégio «Efetuar tradução automática» já não é parte do grupo *Idiomas*.
- «Efetuar tradução automática» está em *Administração* e no novo grupo «Tradução automática».
- Corrigida a geração de ficheiros XLSX com caracteres especiais.
- Adicionada capacidade ao backend de autenticação do GitHub para verificar se o utilizador pertence a uma organização ou equipa específica do GitHub.
- Melhorado o feedback sobre parâmetros inválidos passados ao API.
- Adicionado suporte para tokens de acesso com escopo de projeto para a API.
- Corrigida a remoção de cadeias em alguns casos.
- Corrigida a tradução de cadeias recém-adicionadas.
- Automaticamente etiqueta cadeias traduzidas para facilitar a filtragem delas.

Todas as alterações em detalhes.

4.13 Weblate 4.9.1

Lançado em 19 de novembro de 2021.

- Corrigido o envio de ficheiros monolíngues após a mudança do modelo.
- Melhorado o tratamento de espaços em branco nos marcadores.
- Adicionado suporte para filtrar na API de download.
- Corrigida a exibição de estatísticas ao adicionar novas traduções.
- Mitiga os problemas com a alteração da chave SSH do GitHub.

Todas as alterações em detalhes.

4.14 Weblate 4.9

Lançado em 10 de novembro de 2021.

- Fornece mais detalhes para eventos no histórico.
- Melhorada a renderização de histórico.
- Melhorado o desempenho das páginas de tradução.
- Adicionado suporte para restringir o downloads de ficheiros de tradução.
- O `safe-html` agora pode entender Markdown quando usado com `md-text`.

- A tag `max-length` agora ignora a marcação XML quando usada com `xml-text`.
- Corrigidas as dimensões de textos renderizados em *Tamanho máximo da tradução*.
- Reduzido o comprimento do título da loja de apps para 30 para auxiliar nas próximas alterações da política do Google.
- Adicionado suporte para personalizar a invocação de SSH via `SSH_EXTRA_ARGS`.
- Adicionadas verificações para ICU MessageFormat.
- Melhorado o tratamento de condições de erro em backends de traduções de máquina.
- Realce de caracteres de espaço em branco incomuns nos textos.
- Adicionada opção para permanecer no texto traduzido durante a edição.
- Adicionado suporte para personalizar a invocação de Borg via `BORG_EXTRA_ARGS`.
- Corrigida a geração de ficheiros MO para traduções monolíngues.
- Adicionado endpoint de API para descarregar todas as traduções de componentes como um ficheiro ZIP.
- Suporte adicionado para Pyton 3.10.
- Adicionado suporte para reenviar convite por e-mail a partir da interface de gestão.

Todas as alterações em detalhes.

4.15 Weblate 4.8.1

Lançado em 10 de setembro de 2021.

- Corrigida a remoção de utilizador na interface administrativa do Django.
- Documentados os parâmetros de extensões com mais detalhes.
- Corrigido um erro de JavaScript no glossário.
- Adicionado limite à quantidade de correspondências na verificação de consistência.
- Melhorado o manuseio de espaços reservados em traduções de máquina.
- Corrigida a criação de extensões a usar API.
- Adicionada configuração `PRIVACY_URL` para adicionar uma política de privacidade ao rodapé.
- Oculta os endereços de e-mail dos membros dos administradores do projeto.
- Aprimorada a mesclagem do gettext PO em caso de conflitos.
- Melhorado o realce de glossário.
- Melhorado o comportamento do marcador `safe-html` com verificações de XML.
- Corrigidas as mensagens de componentes vinculados.

Todas as alterações em detalhes.

4.16 Weblate 4.8

Lançado em 21 de agosto de 2021.

- Adicionado suporte para o formato stringsdict da Apple.
- O operador de pesquisa exato agora diferencia maiúsculas de minúsculas com o PostgreSQL.
- Corrigido o salvamento das explicações do glossário em alguns casos.
- Melhorias na documentação.
- Melhorias de desempenho.
- Aprimorada a compatibilidade da extensão de squash com Gerrit.
- Corrigida a adição de textos a componentes monolíngues do glossário.
- Melhorado o desempenho no manuseio de variantes.
- Corrigido o caso de extensão de squash, às vezes, ignorar análise das alterações de upstream.
- Preserva a extensão do ficheiro para descarregas.
- Adicionado suporte para o formato Fluent.
- Adicionado suporte para usar tabulações para formatos de recuo do JSON.

Todas as alterações em detalhes.

4.17 Weblate 4.7.2

Lançado em 15 de julho de 2021.

- Suporta mais apelidos de idioma a serem configurados num projeto.
- Corrigida a validação de texto de pesquisa na API.
- Corrigidas as URLs do exportador Git após uma mudança de domínio.
- Corrigida a extensão de limpeza para ficheiros RC do Windows.
- Corrigida possível falha na atualização de XLIFF.

Todas as alterações em detalhes.

4.18 Weblate 4.7.1

Lançado em 30 de junho de 2021.

- Aprimorado o popup para adicionar termos ao glossário.
- Adicionado suporte para serviço de tradução de máquina do LibreTranslate.
- Adicionada limitação de taxa na criação de novos projetos.
- Melhorado o desempenho de atualizações de ficheiros.

Todas as alterações em detalhes.

4.19 Weblate 4.7

Lançado em 17 de junho de 2021.

- Melhorada a verificação de saúde da configuração.
- Suporte adicionado para `object-pascal-format` usado em gettext PO, veja *Formato Object Pascal*.
- Renomeado *Chaves próximas* para *Chaves similares* para melhor descrever o propósito.
- Adicionado suporte para *Ficheiros lang mi18n*.
- Melhorada a integração de autenticação SAML.
- Corrigida a integração *Gerrit* para melhor lidar com casos complexos.
- Weblate agora necessita de Django 3.2.
- Corrigido o convite de utilizadores quando a autenticação por e-mail está desativada.
- Melhoradas as definições de idioma.
- Adicionado suporte para bloquear utilizadores de contribuir para um projeto.
- Corrigida a criação automática de idiomas de glossário.
- Estendida a documetanção sobre extensões.
- Melhorias no desempenho para componentes com repositórios vinculados.
- Adicionado suporte para DeepL API grátis.
- O utilizador de gerência não precisa mais da interface administrativa do Django.

Todas as alterações em detalhes.

4.20 Weblate 4.6.2

Lançado em 8 de maio de 2021.

- Corrigida a falha após mover o componente compartilhado entre projetos.
- Corrigida a adição de novos textos a ficheiros de propriedades vazios.
- Corrigido o alinhamento de ícone de cópia em idiomas RTL.
- Estendidas as estatísticas de texto na guia de Informações.
- Corrigido o tratamento de ficheiros de tradução ignorados no Git.
- Melhorada o desempenho das métricas.
- Corrigido possível erro no salvamento de glossários.
- Corrigido o comportamento de verificação de consistência em idiomas com diferentes regras plurais.

Todas as alterações em detalhes.

4.21 Weblate 4.6.1

Lançado em 2 de maio de 2021.

- Removido código obsoleto de proteção contra spam.
- Melhora na precisão da verificação de plural de textos fonte.
- Atualizada a lista de idiomas da interface do utilizador no Docker.
- Melhoradas as mensagens de erro ao criar pull requests.
- Corrigida a criação de pull requests no Pagure.
- Corrigido acionamento automático de extensões instaladas.
- Corrigidos possíveis problemas de cache ao atualizar.
- Corrigida a adição de novas unidades para traduções monolíngues usando envio.

Todas as alterações em detalhes.

4.22 Weblate 4.6

Lançado em 19 de abril de 2021.

- O comando de gestão auto_translate agora tem um parâmetro para especificar o modo de tradução.
- Adicionado suporte para *Ficheiros de texto*.
- Adicionadas tendências e métricas para todos os objetos.
- Added support for directly copying text from secondary languages.
- Adicionada filtragem de datas ao navegar pelas alterações.
- Melhorados os gráficos de atividades.
- O remetente para e-mails de formulário de contato agora pode ser configurado.
- Melhorada a validação de parâmetros na API de criação de componentes.
- A limitação de taxa não se aplica mais a superutilizadores.
- Melhorada a desempenho e confiabilidade da extensão de tradução automática.
- A limitação de taxa agora pode ser personalizada no contentor do Docker.
- API para criar componentes agora usa automaticamente *URLs internas do Weblate*.
- Indicação de estado simplificada ao listar textos.
- Hash de palavras-passe agora usa Argon2 por padrão.
- Barras de progresso simplificadas a indicar o estado da tradução.
- Renomeada *Adiciona idiomas em falta* para esclarecer o propósito.
- Corrigido gravar o estado de cadeia para XLIFF.
- Adicionada pesquisa para todo idioma.
- Suporte inicial a *Dimensionando horizontalmente* da implantação do Docker.

Todas as alterações em detalhes.

4.23 Weblate 4.5.3

Lançado em 1º de abril de 2021.

- Corrigida a coleta de métricas.
- Corrigido possível travamento ao adicionar textos.
- Melhorados os exemplos de consulta de pesquisa.
- Corrigida uma possível perda de textos adicionados recentemente no envio de substituição.

4.24 Weblate 4.5.2

Lançado em 26 de março de 2021.

- Agendamento configurável para tradução automática.
- Adicionada verificação de formato Lua.
- Ignorados os textos de formato na verificação *Palavras consecutivas duplicadas*.
- Permitido o envio de uma captura de ecrã de uma página de tradução.
- Adicionada sincronização de ficheiro forçada para a manutenção do repositório.
- Sugestões automáticas corrigidas para idiomas com um código mais longo.
- Melhorado desempenho ao adicionar novos textos.
- Várias correções de erros em verificações de qualidade.
- Várias melhorias de desempenho.
- Adicionada integração com *Descubra Weblate*.
- Corrigido o comportamento das verificações com textos somente leitura.

Todas as alterações em detalhes.

4.25 Weblate 4.5.1

Lançado em 5 de março de 2021.

- Corrigida a edição de marcadores de glossário em alguns casos complexos.
- Estendido o uso de métricas para melhorar o desempenho de várias páginas.
- Armazena o idioma fonte correto em ficheiros TMX.
- Melhor manuseio para envios de PO monolíngues usando API.
- Melhorada o comportamento de alertas em componentes de glossário.
- Melhoradas verificações de ligações Markdown.
- Indicação do glossário e do idioma fonte na estrutura de navegação.
- Listagem paginada de componentes de grandes projetos.
- Aprimorado o desempenho da remoção de tradução, componente ou projeto.
- Melhorado o desempenho de edição em massa.
- Corrigida a preservação dos estados de «Necessita edição» e «Aprovado» para ficheiros ODF.
- Interface melhorada para personalização de descarregas de ficheiros de tradução

Todas as alterações em detalhes.

4.26 Weblate 4.5

Lançado em 19 de fevereiro de 2021.

- Suporte adicionado para `lua-format` usado em gettext PO.
- Adicionado suporte para compartilhar componente entre projetos.
- Corrigido o comportamento de verificação de várias variáveis sem nome com vários marcadores de formato.
- Removido o campo da lista de discussão no projeto em favor de instruções genéricas para tradutores.
- Adicionada extensão de geração de pseudolocalidade.
- Suporte adicionado para ficheiros TermBase eXchange.
- Adicionado suporte para definir manualmente variantes de cadeia a usar um marcador.
- Aprimorado o desempenho de verificações de consistência.
- Aprimorado o desempenho de memória de tradução para textos longos.
- Adicionado suporte para pesquisar em explicações.
- Textos também podem ser adicionados e removidos em formatos bilíngues.
- Estende a lista de idiomas suportados na tradução automática do Amazon Translate.
- Ativa automaticamente verificações de MessageFormat para Properties do Java.
- Adicionado um novo método de envio para adicionar novos textos para uma tradução.
- Adicionada uma interface simples para navegação das traduções.
- Glossários agora são armazenados como componentes normais.
- Removido a API específica para glossários, já que a API de componente é usada agora.
- Adicionada interface simplificada para alternar alguns dos marcadores.
- Adicionado suporte para termos não traduzidos ou proibidos no glossário.
- Adicionado suporte para definir terminologia num glossário.
- Movido o botão da direção do texto para obter mais espaço para o teclado visual.
- Adicionada a opção de observar automaticamente os projetos para os quais o utilizador contribuiu.
- Adicionada a verificação se a tradução corresponde ao glossário.
- Adicionado suporte para personalizar a cor de texto de navegação.

Todas as alterações em detalhes.

4.27 Weblate 4.4.2

Lançado em 14 de janeiro de 2021.

- Corrigida a corrupção de um ficheiro MO distribuído.

4.28 Weblate 4.4.1

Lançado em 13 de janeiro de 2021.

- Corrigida a reversão de alterações plurais.
- Corrigida a exibição de ajuda para as configurações do projeto.
- Administração aprimorada de utilizadores.
- Melhorado o tratamento de contexto em ficheiros PO monolíngues.
- Corrigido o comportamento da extensão de limpeza com os formatos HTML, ODF, IDML e RC do Windows.
- Corrigida a análise do local de ficheiros CSV.
- Uso de compactação de conteúdo para descarregas de ficheiros.
- Melhorada a experiência do utilizador na importação de ficheiro ZIP.
- Melhorada a deteção do formato de ficheiros para envios.
- Evita pull requests duplicadas no Pagure.
- Melhorado o desempenho ao exibir traduções fantasmas.
- Reimplementado o editor de tradução para usar a área de texto do navegador nativo.
- Corrigida a quebra da extensão de limpeza ao adicionar novos textos.
- Adicionada API para extensões.

Todas as alterações em detalhes.

4.29 Weblate 4.4

Lançado em 15 de dezembro de 2020.

- Melhorada a validação ao criar um componente.
- Weblate agora necessita de Django 3.1.
- Adicionado suporte para personalização de aparência na interface de gestão.
- Corrigido o tratamento de estado somente leitura na edição em massa.
- Integração com CodeMirror melhorada.
- Adicionada extensão para remover textos vazios de ficheiros de tradução.
- O editor CodeMirror é agora usado para traduções.
- Destaque de sintaxe no editor de tradução para XML, HTML, Markdown e reStructuredText.
- Realce de objetos colocáveis no editor de tradução.
- Melhorado suporte para códigos de idioma não padrão.
- O componente contém ficheiros de tradução para o idioma fonte.
- O utilizador agora vê uma lista filtrada de idiomas ao adicionar uma nova tradução.
- Estendidos os recursos de pesquisa para mudanças no histórico.
- Melhoradas as páginas de detalhes de cobrança e fluxo de trabalho de hospedagem Livre.
- Estendida a API de estatísticas de tradução.
- Melhorada a guia «outras traduções» ao traduzir.
- Adicionada API de tarefas.

- Melhorado o desempenho do envio de ficheiros.
- Melhorda a exibição de caracteres especiais definidos pelo utilizador.
- Melhorado o desempenho da tradução automática.
- Várias pequenas melhorias na interface do utilizador.
- Melhorada a nomenclatura de downloads ZIP.
- Adicionada a opção para obter notificações de projetos não observados.

  Todas as alterações em detalhes.

4.30 Weblate 4.3.2

Lançado em 4 de novembro de 2020.

- Corrigida a falha em algumas máscaras de ficheiros de componentes.
- Melhorada a precisão de verificação de palavras consecutivas duplicadas.
- Suporte adicional para solicitações de Pagure.
- Melhoradas as mensagens de erro para registos com falha.
- Revertida a renderização de comentários do programador como Markdown.
- Simplificada a configuração de repositórios Git com ramo padrão diferente do «master».
- Repositórios internos recém-criados agora usam main como o ramo padrão.
- Reduzida a taxa de falso-positivos de tradução inalterada durante a tradução de reStructuredText.
- Corrigidos problemas de exibição do CodeMirror em algumas situações.
- Renomeado o grupo de modelos para «Fontes» para esclarecer o significado.
- As pull requests do GitLab em repositórios com caminhos mais longos.

Todas as alterações em detalhes.

4.31 Weblate 4.3.1

Lançado em 21 de outubro de 2020.

- Melhorada o desempenho da tradução automática.
- Expiração da sessão para utilizadores autenticados corrigida.
- Suporte para ocultar informações da versão adicionado.
- Melhoreada a compatibilidade dos ganchos com o Bitbucket Server.
- Melhorado o desempenho de atualizações de memória de tradução.
- Reduzido o uso de memória.
- Melhorado o desempenho da visão Matriz.
- Adicionada confirmação antes de remover um utilizador de um projeto.

  Todas as alterações em detalhes.

4.32 Weblate 4.3

Lançado em 15 de outubro de 2020.

- Incluídas estatísticas de utilizador na API.
- Corrigida a ordem de componentes em páginas paginadas.
- Definido o idioma fonte de um glossário.
- Reescrito o suporte a pull requests do GitHub e GitLab.
- Contagens de estatísticas corrigidas após a remoção da sugestão.
- Perfil do utilizador público estendido.
- Corrigida a configuração de verificações reforçadas.
- Melhorada a documentação sobre backups embutidos.
- Movido o atributo do idioma fonte do projeto para um componente.
- Adicionar a verificação de formatação Vue I18n.
- A verificação de espaços reservados genéricos agora oferece suporte a expressões regulares.
- Melhorado o visual do modo Matriz.
- A maquinaria é agora chamada sugestões automáticas.
- Adicionado suporte para interagir com várias instâncias do GitLab ou GitHub.
- Estendida a API para cobrir atualizações de projeto, atualizações de unidade e remoções e glossários.
- A API de unidade agora lida corretamente com vários textos.
- A criação de componentes agora pode lidar com ficheiros ZIP ou o envio de documentos.
- Códigos de status de resposta API consolidados.
- Suporte a Markdown no acordo de colaborador.
- Rastreamento de cadeias de origem melhorado.
- Melhorada a compatibilidade dos formatos JSON, YAML e CSV.
- Suporte adicional para remover cadeias.
- Melhorado o desempenho de descargas de ficheiros.
- Melhorada a visão de gestão de repositório.
- Ativa automaticamente o formato java para Android.
- Suporte adicional para capturas de ecrã localizadas.
- Suporte adicional para Python 3.9.
- Corrigida a tradução de ficheiros HTML sob certas condições.

Todas as alterações em detalhes.

4.33 Weblate 4.2.2

Lançado em 2 de setembro de 2020.

- Correspondência de cadeias fonte para formatos JSON corrigido.
- Corrigido o redirecionamento de login para algumas configurações de autenticação.
- Autenticação LDAP corrigida com sincronização de grupo.
- Falha na comunicação do progresso da tradução automática corrigida.
- Corrigido o «squashing» de commits do Git com linhas finais ativadas.
- Corrigida a criação de componentes VCS locais a usar API.

4.34 Weblate 4.2.1

Lançado em 21 de agosto de 2020.

- Corrigido o salvamento de plurais para alguns locais em recursos do Android.
- Corrigido o travamento na limpeza da extensão para alguns ficheiros XLIFF.
- Permitida a configuração de CDN de localização na imagem Docker.

4.35 Weblate 4.2

Lançado em 18 de agosto de 2020.

- Melhoradas as páginas de utilizador e adicionada uma listagem de utilizadores.
- Removido o suporte a migração de versões 3.x, migre para 4.1 ou 4.0.
- Adicionadas exportações em vários formatos monolíngues.
- Melhorados os gráficos de atividades.
- A quantidade de textos próximos exibidos pode ser configurado.
- Suporte adicionado para bloquear componentes com erros no repositório.
- Navegação principal simplificada (botões substituídos por ícones).
- Melhorado o manuseio de código de idioma aprimorado na integração com o Google Tradutor.
- A extensão de Git squash pode gerar `Co-authored-by:` ao final da mensagem de commit.
- Melhorado o analisador de pesquisa de consulta.
- Melhorado o feedback de utilizador das verificações de texto de formato.
- Melhorado o desempenho de alterações em massa de estado.
- Adicionados redirecionamentos de compatibilidade após a renomeação do projeto ou componente.
- Adicionadas notificações para aprovação de textos, bloqueio de componentes e alteração de licença.
- Adicionado suporte para ModernMT.
- Permitido evitar sobrescrever traduções aprovadas no envio do ficheiro.
- Removido o suporte para alguns redirecionamentos de URL de compatibilidade.
- Verificação para literais de modelo de ECMAScript adicionada.
- A opção para observar um componente foi adicionada.

- Removido o ponto inicial das chaves da unidade JSON.
- Fila separada de Celery para memória de tradução foi removida.
- Permitir a tradução de todos os componentes de uma só vez.
- Permitida a configuração de cabeçalhos HTTP `Content-Security-Policy`.
- Adicionado suporte para apelidamento de idiomas no nível do projeto.
- Nova extensão para ajudar com localização HTML e JavaScript, veja *CDN de localização JavaScript*.
- O domínio Weblate é agora configurado nas configurações, veja `SITE_DOMAIN`.
- Adicionar suporte para a pesquisa por componente e projeto.

4.36 Weblate 4.1.1

Lançado em 19 de junho de 2020.

- Corrigida a alteração da configuração do autofix ou de extensões no Docker.
- Corrigida uma possível falha na página «Sobre».
- Melhorada a instalação de ficheiros de localidade compilados.
- Corrigida a adição de palavras ao glossário.
- Corrigidos os atalhos de teclado para maquinaria.
- Removida a saída de depuração que causava descarte de eventos de log em algumas instalações.
- Corrigido a indicação de bloqueio na listagem de projeto.
- Corrigida a listagem de chaves GPG em algumas instalações.
- Adicionada opção para qual versão da API do DeepL usar.
- Adicionado suporte para atuar como Provedor de Serviço SAML, veja *Autenticação por SAML*.

4.37 Weblate 4.1

Lançado em 15 de junho de 2020.

- Adicionado suporte para criação de novas traduções com código de país incluído.
- Adicionado suporte para pesquisar cadeias fonte com captura de ecrã.
- Estendidas as informações disponíveis nos insights de estatísticas.
- Melhorada a edição de pesquisa nas páginas «Traduzir».
- Melhorado o manuseio de atualizações de repositório simultâneas.
- Incluir o idioma fonte no formulário de criação do projeto.
- Incluída a contagem de alterações nos créditos.
- Corrigida a seleção de idioma da interface de utilizador em alguns casos.
- Permitida a adição à lista de permissão métodos de registo com registos fechados.
- Melhorada a pesquisa de termos relacionados no glossário.
- Melhoradas as correspondências de memória de tradução.
- Agrupados os mesmos resultados de maquinaria.
- Adicionado uma ligação direta para editar captura de ecrã da página de tradução.

- Melhorada a caixa de diálogo de confirmação de remoção.
- Incluídos modelos no download do ZIP.
- Adicionado suporte para Markdown e configuração de notificação em anúncios.
- Estendidos detalhes em listas de verificações.
- Adicionado suporte para novos formatos de ficheiro: *Cadeias de PHP Laravel*, *Ficheiro HTML*, *Formato OpenDocument*, *Formato IDML*, *Ficheiros RC do Windows*, *Traduções de INI*, *Traduções Inno Setup INI*, *Propriedades GWT*, *Ficheiros JSON go-i18n*, *Ficheiro ARB*.
- Uso consistentemente de dispensado como um estado de verificações dispensadas.
- Adiciona suporte para configurar extensões predefinidas para ativar.
- Corrigido o atalho de teclado do editor para dispensar verificações.
- Melhorada a tradução de máquina de textos com espaços reservados.
- Mostrada a tradução fantasma para os idiomas do utilizador para facilitar o início deles.
- Melhorada a análise de código de idioma.
- Mostra as traduções no idioma do utilizador primeiro na lista.
- Renomeados os formatos para variantes de nome mais genéricas.
- Adicionadas novas verificações de qualidade: *Várias variáveis sem nome*, *Não traduzido há muito tempo*, *Palavras consecutivas duplicadas*.
- Reintroduzido o suporte a limpar a memória de tradução.
- Corrigida a opção para ignorar verificações em texto fonte.
- Adicionado suporte para configurar diferentes ramos para fazer push de alterações.
- A API agora relata status da limitação de taxa nos cabeçalhos HTTP.
- Adicionado suporte para API V3 do Google Tradutor (avançado).
- Adicionada capacidade de restringir o acesso no nível do componente.
- Adicionado suporte para espaços em branco e outros caracteres especiais nos sinalizadores de tradução, veja *Personalizando comportamento a usar marcadores*.
- Sempre mostra a verificação de texto renderizado se ativada.
- A API agora tem suporta a filtragem de alterações.
- Adicionado suporte para compartilhar glossários entre projetos.

4.38 Weblate 4.0.4

Lançado em 7 de maio de 2020.

- Corrigida a execução do conjunto de testes em alguns ambientes Python 3.8.
- Correções de erro ortográficos na documentação.
- Corrigida a criação de componentes a usar API em alguns casos.
- Corrigidos erros de JavaScript que interrompiam a navegação móvel.
- Corrigida a falha na exibição de algumas verificações.
- Corrigida a listagem de capturas de ecrã.
- Corrigidas notificações de resumo mensais.
- Corrigido o comportamento de tradução intermediário com unidades não existentes na tradução.

4.39 Weblate 4.0.3

Lançado em 2 de maio de 2020.

- Corrigida possível falha em relatórios.
- As menções do utilizador nos comentários agora não diferenciam maiúsculas de minúsculas.
- Corrigida a migração do PostgreSQL para não superutilizadores.
- Corrigida a alteração da URL do repositório durante a criação do componente.
- Corrigida uma falha quando o repositório upstream desaparece.

4.40 Weblate 4.0.2

Lançado em 27 de abril de 2020.

- Melhorado o desempenho das estatísticas de tradução.
- Melhorado o desempenho de alterações de rótulos.
- Melhorado o desempenho de edição em massa.
- Melhoria do desempenho da memória de tradução.
- Corrigida possível falha na exclusão do componente.
- Corrigida a exibição de alterações de tradução em alguns casos complexos.
- Melhorado o aviso sobre fila muito longa do Celery.
- Corrigidos possíveis falsos positivos na verificação de consistência.
- Corrigido um impasse ao alterar o repositório de componentes vinculados.
- Incluída a distância de edição na lista de alterações e CSV e relatórios.
- Evita falsos positivos de verificação de espaçamento de pontuação para o francês canadense.
- Corrigida a exportação XLIFF com espaços reservados.
- Corrigido falso-positivo com verificação de largura zero.
- Melhorado o relatório de erros de configuração.
- Corrigido o envio de fonte bilíngue.
- Detecta automaticamente os idiomas suportados para tradução de máquina DeepL.
- Corrigida a exibição da barra de progresso em alguns casos complexos.
- Corrigidas algumas verificações acionadas em textos não traduzidos.

4.41 Weblate 4.0.1

Lançado em 16 de abril de 2020.

- Corrigida a instalação de pacotes do PyPI.

4.42 Weblate 4.0

Lançado em 16 de abril de 2020.

- O Weblate agora requer Python 3.6 ou mais recente.
- Adicionada visão geral do gestão de alertas de componentes.
- Adicionado alerta de componente para URLs quebradas do navegador do repositório.
- Melhoradas páginas de autenticação e registo.
- Controle de acesso ao projeto e configuração do fluxo de trabalho integrado às configurações do projeto.
- Adicionados verificação e realce para interpolação e aninhamento i18next.
- Adicionados verificação e marcador para espaços reservados de percentagem.
- Mostrar falhas nas verificações de sugestões.
- Regista as alterações de texto fonte no histórico.
- Atualizado o Microsoft Translator à versão 3 da API.
- Reimplementado backend da memória de tradução.
- Adicionado suporte a várias pesquisa `is:` em *Pesquisando*.
- Permite fazer *Tradução inalterada* evitar lista negra interna.
- Melhorada a extração de comentários de ficheiros po monolíngues.
- Renomeadas as mensagens do quadro de comunicação para anúncios.
- Corrigidos problemas ocasionais com e-mails de registo.
- Melhorada a extensão de atualização do LINGUAS para lidar com mais variantes de sintaxe.
- Corrigida a edição do ficheiro fonte XLIFF monolíngue.
- Adicionado suporte para correspondência exata em *Pesquisando*.
- Estendida a API para cobrir capturas de ecrã, utilizadores, grupos, listas de componentes e projetos de criação estendida.
- Adicionado suporte para envio de fonte em traduções bilíngues.
- Adicionado suporte para idioma intermediário de programadores.
- Adicionado suporte para revisão de textos fonte.
- Estendidas as opções de download para memória de tradução em toda a plataforma.

4.43 Série 3.x do Weblate

4.43.1 Weblate 3.11.3

Lançado em 11 de março de 2020.

- Corrigida a pesquisa de campos com certa prioridade.
- Corrigida a consulta predefinida para textos adicionados recentemente.
- Corrigida a pesquisa que retornava correspondências duplicadas.
- Corrigido o processamento de notificações no Gmail.
- Corrigida a reversão das alterações do histórico.
- Adicionados ligações para eventos nas notificações de resumo.

- Corrigido o e-mail para confirmação de remoção de conta.
- Adicionado suporte para autenticação Slack no contentor Docker.
- Evita enviar notificações para idiomas não assinados.
- Inclusão de filas do Celery na visão geral de desempenho.
- Corrigidos os links da documentação para extensões.
- Redução de falso-negativos para verificação de tradução inalterada.
- Elevada a dependência do bleach para abordar CVE-2020-6802.
- Corrigida a lista de mudanças de nível de projeto no histórico.
- Corrigida a invalidação de estatísticas em alguns casos complexos.
- Corrigida a pesquisa de determinados estados de texto.
- Corrigido o comportamento de verificação de formato de texto em caso de faltar percentagem.
- Corrigida a autenticação a usar alguns provedores de terceiros.

4.43.2 Weblate 3.11.2

Lançado em 22 de fevereiro de 2020.

- Corrigida a renderização de sugestões.
- Corrigidas algumas strings relatadas incorretamente como não tendo palavras.

4.43.3 Weblate 3.11.1

Lançado em 20 de fevereiro de 2020.

- Documentadas as alterações de configuração no Celery.
- Melhorada a validação de nome de ficheiros na criação de componentes.
- Corrigidas as versões mínimas de algumas dependências.
- Corrigida a adição de grupos com certas versões do Django.
- Corrigido o push manual para o repositório upstream.
- Melhorada a correspondência de glossário.

4.43.4 Weblate 3.11

Lançado em 17 de fevereiro de 2020.

- Permite o uso de URL de push VCS durante a criação do componente via API.
- A verificação de largura renderizada agora mostra a imagem com a renderização.
- Corrigidas as ligações em e-mails de notificação.
- Melhorada a aparência de e-mails em texto simples.
- Exibe verificações ignoradas e permite torná-las ativas novamente.
- Exibe as teclas próximas em traduções monolíngues.
- Suporte adicionado para agrupar formas de cadeias.
- Recomenda a atualização para novas versões do Weblate nas verificações do sistema.
- Fornece uma análise mais detalhada para alerta de idioma duplicado.

- Inclui informações mais detalhadas sobre a licença nas páginas do projeto.
- Automaticamente desfaz shallow em cópias locais, se necessário.
- Consertado o download de textos a precisar de ação.
- Novo alerta para alertar sobre o uso da mesma máscara de ficheiros duas vezes.
- Melhorar a extração de objetos colocáveis XML.
- O `SINGLE_PROJECT` agora pode forçar o redirecionamento para o projeto escolhido.
- Adicionada opção para resolver comentários.
- Adicionada edição em massa de marcadores.
- Adicionado suporte para labels.
- Adicionada extensão de edição em massa.
- Adicionada opção para *Forçar verificações*.
- Aumento da validade padrão das ligações de confirmação.
- Melhorada integração com o Matomo.
- Corrigido *Foi traduzido* para tratar corretamente da mudança do texto fonte.
- Configuração estendida de atualizações automáticas por `AUTO_UPDATE`.
- Extensões de LINGUAS agora fazem sincronização completa de tradução no Weblate.

4.43.5 Weblate 3.10.3

Lançado em 18 de janeiro de 2020.

- Suporte para translate-toolkit 2.5.0.

4.43.6 Weblate 3.10.2

Lançado em 18 de janeiro de 2020.

- Adicionar indicação de bloqueio aos projetos.
- Corrigido erro CSS que causava cintilação em alguns navegadores web.
- Corrigida a pesquisa em sistemas com localidades diferentes do inglês.
- Melhorada correspondência de repositório para hooks do GitHub e do Bitbucket.
- Corrigida migração de dados em algumas instalações do Python 2.7.
- Permitir a configuração da clonagem rasa de Git.
- Melhorado processamento de notificações de segundo plano.
- Corrigido o envio de formulários quebrados ao navegar de volta no navegador da web.
- Nova extensão para configurar a formatação YAML.
- Corrigida a verificação de mesmos plurais para não disparar em idiomas de forma plural única.
- Corrigida pesquisa por expressões regulares em alguns campos.

4.43.7 Weblate 3.10.1

Lançado em 9 de janeiro de 2020.

- API estendida com criação de tradução.
- Corrigidos diversos casos especiais em migrações de dados.
- Compatibilidade com Django 3.0.
- Aprimorado o desempenho da limpeza de dados.
- Adicionado suporte ao security.txt customizável.
- Melhorado a estrutura de navegação no changelog.
- Melhorada listagem de traduções no painel.
- Melhorada respostas HTTP para webhooks.
- Adicionado suporte aos merge requests do GitLab no contentor Docker.

4.43.8 Weblate 3.10

Lançado em 20 de dezembro de 2019.

- Interface de utilizador da aplicação melhorada.
- Adicionada checagem de duplos espaços.
- Consertada criação de novos idiomas.
- Evita enviar notificações de log de auditoria para e-mails excluídos.
- Adicionado suporte à textos somente leitura.
- Adicionado suporte a Markdown nos comentários.
- Permitir posicionamento de textos de instruções de tradução nas informações do projeto.
- Adicionar cópia para área de transferência para idiomas secundários.
- Melhorado suporte ao Mercurial.
- Melhorado desempenho de busca do repositório Git.
- Adicionada pesquisa pela idade do texto.
- Mostrar idioma fonte para todas as traduções.
- Mostrar contexto para textos próximos.
- Adicionado suporte para notificações nas operações do repositório.
- Melhoradas listagens de tradução.
- Capacidades de pesquisa estendidos.
- Adicionado suporte para tradução automática de textos marcados para edição.
- Evitar o envio de notificações duplicadas para alertas de componentes vinculados.
- Melhorada a mensagem da merge request padrão.
- Melhor indicação do estado do texto no modo Zen.
- Adicionado suporte a mais idiomas no Yandex Translate.
- Visual melhorado de notificações de e-mails.
- Fornecer seleção para a licença de tradução.

4.43.9 Weblate 3.9.1

Lançado em 28 de outubro de 2019.

- Remover alguns ficheiros desnecessários dos backups.
- Corrigida possível falha nos relatórios.
- Corrigida uma falha de migração cruzada de bancos de dados.
- Adicionado suporte para forçar o push de repositórios Git.
- Reduzido o risco de invalidação de token de registo.
- Corrigido o limitador de taxa de acerto de remoção de conta.
- Adicionada pesquisa baseada em prioridade.
- Consertada possível falha ao adicionar textos para um ficheiro JSON.
- A verificação e correção de HTML seguro agora respeitam a marcação do texto fonte.
- Impedir o envio de notificações para utilizadores convidados e deletados.
- Consertar a conexão SSL para o redis no Celery num contentor Docker.

4.43.10 Weblate 3.9

Lançado em 15 de outubro de 2019.

- Incluir metadados do Weblate em ficheiros descarregados.
- Melhorada IU para verificações com falha.
- Indica textos ausentes nas verificações de formatação.
- Verificação separada para o espaçamento de pontuação francês.
- Adicionado suporte para consertar alguns erros de verificação de qualidade.
- Adicionada permissão separada para criar novos projetos.
- Estender estatísticas para a contagem de caracteres.
- Melhorado o suporte aos códigos de idioma no estilo Java.
- Adicionada nova verificação genérica para espaços reservados.
- Adicionado suporte para espaços reservados de WebExtension JSON.
- Adicionado suporte para formato XML simples.
- API estendida com remoção e criação de tradução de projeto, componente e tradução.
- Adicionado suporte para webhooks do Gitea e Gitee.
- Adicionada nova verificação de expressões regulares personalizadas.
- Permitida a configuração de contribuições para a memória de tradução compartilhada.
- Adicionado a descarrega de ZIP para mais ficheiros de tradução.
- Tornada a análise compatível com o padrão XLIFF de altura máxima e fonte.
- Adicionados uma nova verificação e correção para marcação HTML segura para traduzir aplicações web.
- Adicionado um alerta de componente em configuração sem suporte.
- Adicionada uma extensão de tradução automática para inicializar traduções.
- Estender tradução automática para adicionar sugestões.
- Exibir parâmetros adicionais na visão geral.

- Sentry agora é compatível com o moderno Sentry SDK em vez do Raven.
- Alteradas as configurações de exemplo para se adequarem melhor ao ambiente de produção.
- Adicionados backups automatizados a usar BorgBackup.
- Divida a extensão de limpeza para RESX para evitar atualizações indesejadas de ficheiros.
- Adicionados recursos de pesquisa avançada.
- Permitido que os utilizadores decarregarem os próprios relatórios.
- Adicionado guia de localização para ajudar na configuração de componentes.
- Adicionado suporte para merge requests do GitLab.
- Melhorada a exibição do status do repositório.
- Efetua a tradução de máquina em segundo plano.

4.43.11 Weblate 3.8

Lançado em 15 de agosto de 2019.

- Adicionado suporte para criação simplificada de componentes semelhantes.
- Adicionado suporte para analisar sinalizadores de tradução de formatos de ficheiros baseados em XML.
- Regista as exceções no log do Celery.
- Melhoria no desempenho de extensões com escopo de repositório.
- Visual melhorado de notificações de e-mails.
- Correção do comportamento de redefinição de palavra-passe.
- Corrigido o desempenho na maioria das páginas de tradução.
- Corrigida a lista de idiomas não conhecidos pelo Weblate.
- Adiciona suporte para clonar extensões para componentes descobertos.
- Adiciona suporte para substituir o conteúdo do ficheiro com o enviado.
- Adicionado suporte para tradução de conteúdo não baseado em VCS.
- Adicionada a imagem de widget OpenGraph para uso em redes sociais.
- Adicionado suporte para capturas de ecrã animadas.
- Melhorado o manuseio de ficheiros XLIFF monolíngues.
- Evita enviar várias notificações para um único evento.
- Adicionado suporte para filtrar alterações.
- Estendidos os períodos predefinidos para relatórios.
- Adicionado suporte de webhook para Azure Repos.
- Novas notificações opcionais sobre sugestões pendentes ou textos não traduzidos.
- Adicionado uma ligação de cancelamento de assinatura, por meio de um clique único, aos e-mails de notificação.
- Corrigidos falsos positivos com a verificação «Foi traduzido».
- Nova interface de gestão para administradores.
- A prioridade do texto agora pode ser especificada a usar marcadores.
- Adicionadas visões de gestão de idioma.
- Adicionadas verificações da bibliotecas Qt e textos de formato Ruby.

- Adicionada configuração para melhor se adequar às instalações de um único projeto.
- Notificação sobre novo texto na mudança do texto fonte em traduções monolíngues.
- Adicionada visão separada para memória de tradução com capacidade de pesquisa.

4.43.12 Weblate 3.7.1

Lançado em 28 de junho de 2019.

- Atualizações de documentação.
- Corrigidas algumas restrições de requisitos.
- Atualizado o banco de dados do idioma.
- Atualizações de localização.
- Vários ajustes da interface do utilizador.
- Melhorado o tratamento de ficheiros de tradução não suportados, mas descobertos.
- Relata mais detalhadamente os requisitos de formato de ficheiros ausentes.

4.43.13 Weblate 3.7

Lançado em 21 de junho de 2019.

- Adicionada fila Celery separada para notificações.
- Usa uma aparência consistente com a aplicação para a navegação na API.
- Incluir estatísticas aprovadas nos relatórios.
- Reportar progresso quando atualizar um componente de tradução.
- Permite abortar a atualização do componente em segundo plano.
- Estendida a linguagem do modelo para manipulações de nome de ficheiro.
- Usa modelos para a ligação do editor e URL do navegador do repositório.
- Indica o comprimento máximo e a contagem atual de caracteres ao editar a tradução.
- Melhorado o tratamento de abreviações na verificação de tradução inalterada.
- Página de destino atualizada para novos colaboradores.
- Adiciona suporte para configurar extensões de msgmerge.
- Atraso na abertura da conexão SMTP ao enviar notificações.
- Melhorado o registo de erros.
- Permite local personalizado na extensão de geração de MO.
- Adicionadas extensões para limpar sugestões ou comentários antigos.
- Adicionada a opção para ativar o modo horizontal no editor Zen.
- Melhorado o desempenho de importação com muitos componentes vinculados.
- Corrigida a instalação de exemplos em alguns casos.
- Melhorada a renderização de alertas em mudanças.
- Adicionado novo widget de estatísticas horizontais.
- Melhorada a verificação de textos de formato em plurais.
- Adicionada ferramenta de gestão de fontes.

- Nova verificação das dimensões do texto renderizado.
- Adicionado suporte para formatos de legenda.
- Inclui estatísticas gerais de conclusão para idiomas.
- Adicionados relatórios no escopo global e no de projeto.
- Melhorada a interface de utilizador ao mostrar o status da tradução.
- Novo logotipo e esquema de cores do Weblate.
- Nova aparência de emblemas de bitmap.

4.43.14 Weblate 3.6.1

Lançado em 26 de abril de 2019.

- Melhorado o manuseio de ficheiros XLIFF monolíngues.
- Corrigidas as notificações de resumo em alguns casos complexos.
- Corrigido um alerta de erro de script de extensão.
- Corrigido o ficheiro de geração de MO para ficheiros PO monolíngues.
- Corrigida a exibição de verificações desinstaladas.
- Indica projetos administrados na lista de projetos.
- Permite que a atualização se recupere de um repositório VCS ausente.

4.43.15 Weblate 3.6

Lançado em 20 de abril de 2019.

- Adiciona suporte para descarregar dados do utilizador.
- Extensões agora são acionados automaticamente após a instalação.
- Melhordas as instruções para resolver conflitos de mesclagem.
- A extensão de limpeza agora é compatível com traduções de metadados da loja de apps.
- Sintaxe do código de idioma configurável ao adicionar novas traduções.
- Avisa sobre o uso de Python 2 com encerramento planejado do suporte em abril de 2020.
- Extrai caracteres especiais do texto fonte para o teclado visual.
- Estendidas as estatísticas de colaboradores para refletir as contagens fonte e destino.
- Os administradores e extensões de consistência agora podem adicionar traduções, mesmo se desativado para os utilizadores.
- Corrigida a descrição de alternar a desativar a manipulação do cabeçalho `Language-Team`.
- Notifica os utilizadores mencionados nos comentários.
- Removida a detecção automática do formato de ficheiros da configuração do componente.
- Corrigido o ficheiro de geração de MO para ficheiros PO monolíngues.
- Adicionadas notificações de resumo.
- Adicionado suporte para silenciar notificações de componentes.
- Adicionadas notificações para novos alertas, mensagens do quadro de comunicações ou componentes.
- As notificações para projetos administrados agora podem ser configuradas.
- Melhorado o tratamento de códigos de idioma de três letras.

4.43.16 Weblate 3.5.1

Lançado em 10 de março de 2019.

- Corrigido o exemplo da unidade do systemd do Celery.
- Corrigidas as notificações de repositórios HTTP com login.
- Corrigida a condição de corrida na edição de texto fonte para traduções monolíngues.
- Inclui a saída da execução da extensão com falha nos logs.
- Melhorada a validação de opções para adicionar novo idioma.
- Permite editar o formato do fircheiros nas configurações do componente.
- Atualiza as instruções de instalação para preferir o Python 3.
- Melhorias de desempenho e consistência para carregar traduções.
- Torne o serviço de Terminologia da Microsoft compatível com as versões atuais da Zeep.
- Atualizações de localização.

4.43.17 Weblate 3.5

Lançado em 3 de março de 2019.

- Melhorado o desempenho da memória de tradução embutida.
- Adicionada interface para gerir a memória de tradução global.
- Melhorado o alerta sobre estado inválido do componente.
- Adicionada interface de utilizador para gerir mensagens do quadro de comunicações.
- A mensagem de comit da extensão agora pode ser configurada.
- Reduz a quanidade de commits ao atualizar o repositório upstream.
- Corrigida a possível perda de metadados ao mover um componente entre projetos.
- Melhorada a navegação no modo Zen.
- Adicionadas várias novas verificações de qualidade (relacionadas a Markdown e URL).
- Adicionado suporte para ficheiro de metadados de loja de apps.
- Adicionado suporte para alternar a integração GitHub ou Gerrit.
- Adicionada verificação de letras Kashida.
- Adicionada opção para fazer squash de commits com base nos autores.
- Melhorado o suporte ao formato de ficheiros XLSX.
- Compatibilidade com Tesseract 4.0.
- A extensão de cobrança agora remove projetos para cobranças não pagas após 45 dias.

4.43.18 Weblate 3.4

Lançado em 22 de janeiro de 2019.

- Adicionado suporte para espaços reservados do XLIFF.
- O Celery agora pode utilizar várias filas de tarefas.
- Adicionado suporte para renomear e mover projetos e componentes.
- Inclui contagens de caracteres em relatórios.
- Adicionada adição guiada de componentes de tradução com detecção automática de ficheiros de tradução.
- Mensagens de commit de mesclagem personalizáveis para Git.
- Adicionada indicação visual de alertas de componentes na navegação.
- Melhorado o dsempenho de carregamento de ficheiros de tradução.
- Nova extensão para fazer squash de commits antes de fazer push.
- Melhorada a exibição de alterações de tradução.
- Alterado o estilo de mesclagem padrão ao fazer rebase e tornado-o configurável.
- Melhor tratamento de subtags de uso privado no código de idioma.
- Melhorado o desempenho de atualizações de índice de texto completo.
- Estendida a API de envio de ficheiros para oferecer suporte a mais parâmetros.

4.43.19 Weblate 3.3

Lançado em 30 de novembro de 2018.

- Adicionado suporte para remoção de componentes e projetos.
- Melhorado o desempenho para algumas traduções monolíngues.
- Adicionados alertas de componente de tradução para destacar problemas com uma tradução.
- Expõe a «resname» do XLIFF como contexto quando disponível.
- Adicionado suporte para estados XLIFF.
- Adicionada verificação de ficheiros que não podem ser escritos dentro do DATA_DIR.
- Melhorada a exportação de CSV para alterações.

4.43.20 Weblate 3.2.2

Lançado em 20 de outubro de 2018.

- Remova a dependência do Babel que já não é necessária.
- Definições de idioma atualizadas.
- Melhore a documentação para complementos, LDAP e Aipo.
- Corrigindo a ativação de novos sinaliyadores dos-eol e auto-java-messageformat.
- Corrigido o teste setup.py em execução do pacote PyPI.
- Melhor manuseio de plurais.
- Corrigida a falha de API ao envio de traduções em alguns casos limítrofes.
- Corrigida a atualização da configuração do Git no caso de ser alterada manualmente.

4.43.21 Weblate 3.2.1

Lançado em 10 de outubro de 2018.

- Dependência de documentos em backports.csv em Python 2.7.
- Corrigidos testes executados pelo root.
- Melhor manuseio de erros no módulo gitexport.
- Relatório de progresso corrigido para idiomas recém-adicionados.
- Comunica corretamente os erros do worker do Celery ao Sentry.
- Corrigido criar traduções com o Qt Linguist.
- Corrigidas falhas ocasionais de atualização de índice de texto completo.
- Validação melhorada ao criar componentes.
- Adicionado suporte para a limpeza de sugestões antigas.

4.43.22 Weblate 3.2

Lançado em 6 de outubro de 2018.

- Adicionar comando de gestão install_addon para a instalação automática de extensões.
- Permitir configurações de limites de taxa com uma proporção mais refinada.
- Adicionado suporte para a exportação e importação de ficheiros Excel.
- Melhorar a limpeza dos componentes no caso de múltiplas extensões de descoberta de componentes.
- Reescrito o back-end de tradução automática de Terminologia da Microsoft.
- Weblate agora usa Celery para descarregar algum processamento.
- Melhorados os recursos de pesquisa e adicionada a pesquisa de expressão regular.
- Adicionado suporte para tradução de máquina da API Youdao Zhiyun.
- Adicionado suporte para tradução máquina da API do Baidu.
- Tarefas integradas de manutenção e limpeza usando o Celery.
- Melhor desempenho de carregamento de traduções em quase 25%.
- Removido o suporte para mesclar cabeçalhos no envio.
- Removido o suporte para mensagens de commit personalizadas.
- Modo de edição configurável (zen/full).
- Adicionado suporte para relatórios de erros para o Sentry.
- Adicionado suporte para atualização diária automatizada de repositórios.
- Adicionado suporte para a criação de projetos e componentes por utilizadores.
- Memória de tradução embutida agora armazena automaticamente as traduções feitas.
- Utilizadores e projetos podem importar as memórias de tradução existentes deles.
- Melhor gestão de textos relacionados às capturas de ecrã.
- Adicionado suporte para verificar Java MessageFormat.

Veja o marco 3.2 no GitHub para uma lista detalhada dos problemas resolvidos.

4.43.23 Weblate 3.1.1

Lançado em 27 de julho de 2018.

- Corrigida a falha do conjunto de teste em algumas configurações.

4.43.24 Weblate 3.1

Lançado em 27 de julho de 2018.

- Atualizações de versões anteriores a 3.0.1 não são suportadas.
- Permite substituir mensagens de commit padrão das configurações.
- Melhora a compatibilidade dos webhooks com ambientes de hospedagem própria.
- Adicionado suporte para o Amazon Translate.
- Compatibilidade com Django 2.1.
- As verificações do sistema Django agora são usadas para diagnosticar problemas com a instalação.
- Removido o suporte para o serviço libravatar que será encerrado em breve.
- Nova extensão para marcar traduções não alteradas como a necessitar de edição.
- Adiciona suporte para pular para um local específico durante a tradução.
- As traduções descarregadas podem ser personalizadas agora.
- Melhorado o cálculo de similaridade de texto em correspondências de memória de tradução.
- Adicionado suporte para assinar commits do Git pelo GnuPG.

4.43.25 Weblate 3.0.1

Lançado em 10 de junho de 2018.

- Corrigido um possível problema de migração do 2.20.
- Atualizações de localização.
- Removidos exemplos de ganchos obsoletos.
- Documentação de cache aprimorada.
- Exibição corrigida da documentação do administrador.
- Tratamento aprimorado de nomes longos do idioma.

4.43.26 Weblate 3.0

Lançado em 1º de junho de 2018.

- Controle de acesso reescrito.
- Várias limpezas de código que levam a módulos movidos e renomeados.
- Nova extensão para descoberta automática de componentes.
- O comando de gestão import_project tem parâmetros ligeiramente diferentes agora.
- Adicionado suporte básico para ficheiros de Windows RC.
- Nova extensão para armazenar nomes de contribuidores em cabeçalhos de ficheiros de PO.
- Os scripts de gancho por componente foram removidos; em vez disso, use extensões.
- Adiciona suporte para coletar acordos de colaborador.

- As alterações de controle de acesso agora são rastreadas no histórico.
- Nova extensão para garantir que todos os componentes num projeto tenham as mesmas traduções.
- Suporte para mais variáveis em modelos de mensagem de commit.
- Adiciona suporte para fornecer contexto textual adicional.

4.44 Série 2.x do Weblate

4.44.1 Weblate 2.20

Lançado em 4 de abril de 2018.

- Melhorada a velocidade de clonagem de repositórios de subversion.
- Alterado o bloqueio de repositório para usar biblioteca de terceiros.
- Adicionado suporte para descarregar apenas textos que precisam de ação.
- Adicionado suporte para pesquisa em vários idiomas ao mesmo tempo.
- Nova extensão para configurar a quebra de linha da saída gettext.
- Nova extensão para configurar a formatação JSON.
- Adicionado suporte para autenticação em API usando autenticação Bearer compatível com RFC 6750.
- Adicionado suporte para tradução automática usando serviços de tradução de máquina.
- Adicionado suporte para marcação HTML em mensagens de quadro de comunicações.
- Adicionado suporte para alteração em massa do estado dos textos.
- O translate-toolkit pelo menos 2.3.0 agora é necessário, as versões mais antigas não são mais suportadas.
- Adicionada memória de tradução embutida.
- Adicionada uma visão geral com as listas de componentes ao painel e páginas de visão geral de cada lista de componentes.
- Adicionado suporte para serviço de tradução de máquina DeepL.
- Os resultados da tradução de máquina agora são armazenados em cache no Weblate.
- Adicionado suporte para reordenar alterações de commits feitos.

4.44.2 Weblate 2.19.1

Lançado em 20 de fevereiro de 2018.

- Corrigido o problema de migração na atualização do 2.18.
- Melhorada a validação de API de envio de ficheiros.

4.44.3 Weblate 2.19

Lançado em 15 de fevereiro de 2018.

- Corrigidas as importações em alguns formatos de ficheiros.
- Exibe informações de navegador amigáveis para humanos no registo de auditoria.
- Adicionado um exportador TMX para ficheiros.
- Várias melhorias de desempenho para carregar ficheiros de tradução.
- Adicionada opção para desativar a gestão de acesso no Weblate em favor da gestão do Django.
- Melhorada a velocidade de pesquisa de glossário para textos grandes.
- Compatibilidade com django_auth_ldap 1.3.0.
- Os erros de configuração agora são armazenados e relatados de forma persistente.
- Honra ignorar marcadores na correção automática de espaço em branco.
- Melhorada a compatibilidade com algumas configurações do Subversion.
- Melhorado o serviço embutido de tradução de máquina.
- Adicionado suporte para serviço SAP Translation Hub.
- Adicionado suporte para serviço de Terminologia da Microsoft.
- Removido o suporte para anúncio em e-mails de notificação.
- Melhorado o relatório de progresso de tradução a nível de idioma.
- Melhorado o suporte para diferentes fórmulas plurais.
- Adicionado suporte para repositórios Subversion que não usam stdlayout.
- Adicionadas extensões para personalizar fluxos de trabalho de tradução.

4.44.4 Weblate 2.18

Lançado em 15 de dezembro de 2017.

- Estatísticas estendidas de colaborador.
- Melhor configuração do teclado virtual de caracteres especiais.
- Adicionado suporte para formato de ficheiros de DTD.
- Alterados os atalhos do teclado para reduzir a chance de colisão com os do navegador/sistema.
- Melhorado o suporte para marcador de aprovado em ficheiros de XLIFF.
- Adicionado suporte para não agrupar textos longos em ficheiros de PO gettext.
- Adicionado botão para copiar a ligação permanente à tradução atual.
- Removido o suporte ao Django 1.10 e adicionado suporte ao Django 2.0.
- Removido o bloqueio de traduções durante a tradução.
- Adicionado suporte para adicionar novos textos a traduções monolíngues.
- Adicionado suporte para fluxos de trabalho de tradução com revisores dedicados.

4.44.5 Weblate 2.17.1

Lançado em 13 de outubro de 2017.

- Corrigido o conjunto de testes em execução em algumas situações específicas.
- Atualizações de localidades.

4.44.6 Weblate 2.17

Lançado em 13 de outubro de 2017.

- O Weblate por padrão agora faz clones rasos do Git.
- Melhorado o desempenho ao atualizar ficheiros de tradução grandes.
- Adicionado suporte para bloquear o registo de certos e-mails.
- Os utilizadores agora podem apagar os próprios comentários.
- Adicionada etapa de visualização para o recurso de pesquisar e substituir.
- Persistência das configurações do lado do cliente em formulários de pesquisa e envio.
- Capacidades de pesquisa estendidos.
- Mais granularidade por configuração de ACL do projeto.
- O valor padrão de BASE_DIR foi alterado.
- Adicionada a remoção da conta em duas etapas para evitar a remoção acidental.
- As configurações de controle de acesso do projeto agora podem ser editadas.
- Adicionada proteção opcional contra spam para sugestões usando Akismet.

4.44.7 Weblate 2.16

Lançado em 11 de agosto de 2017.

- Várias melhorias de desempenho.
- Adicionado suporte para o formato JSON aninhado.
- Adicionado suporte para o formato WebExtension JSON.
- Corrigida a autenticação com exportador git.
- Melhorada a importação de CSV em certas situações.
- Melhorada a aparência do widget Outras traduções.
- As verificações de comprimento máximo agora estão impondo o comprimento do texto no formulário.
- Torna a idade de commit_pending configurável por componente.
- Várias limpezas de interface do utilizador.
- Corrigida a pesquisa em componente /projeto/tido o site por traduções.

4.44.8 Weblate 2.15

Lançado em 30 de junho de 2017.

- Mostra mais traduções relacionadas em outras traduções.
- Adiciona a opção de ver as traduções do texto atual em outros idiomas.
- Usa 4 formas plurais para lituano por padrão.
- Corrigido o envio para ficheiros monolíngues de formato diferente.
- Melhoradas as mensagens de erro na autenticação com falha.
- Mantelém o estado da página ao remover a palavra do glossário.
- Adicionada ligação direta para editar a tradução do idioma secundário.
- Adicionada verificação de qualidade do formato Perl.
- Adicionado suporte para rejeitar palavras-passe reutilizadas.
- Estendida a barra de ferramentas para edição de idiomas RTL.

4.44.9 Weblate 2.14.1

Lançado em 24 de maio de 2017.

- Corrigido um possível erro ao paginar os resultados da pesquisa.
- Corrigidas as migrações de versões mais antigas em alguns casos complexos.
- Corrigido possível CSRF ao observar e deixar de observar um projeto.
- A redefinição de palavra-passe não autentica mais o utilizador.
- Corrigido um possível desvio de CAPTCHA em caso de palavra-passe esquecida.

4.44.10 Weblate 2.14

Lançado em 17 de maio de 2017.

- Adiciona entradas de glossário usando AJAX.
- O logout agora usa POST para evitar CSRF.
- A redefinição do token da chave de API usa agora POST para evitar CSRF.
- Weblate define Content-Security-Policy por padrão.
- A URL do editor local é validada para evitar auto-XSS.
- A palavra-passe agora é validada contra falhas comuns por padrão.
- Notificar os utilizadores sobre atividades importantes nas contas deles, como alteração de palavra-passe.
- As exportações de CSV agora escapam de fórmulas potenciais.
- Várias pequenas melhorias na segurança.
- As tentativas de autenticação agora têm taxa limitada.
- O conteúdo da sugestão é armazenado no histórico.
- Armazena atividades importantes da conta no registo de auditoria.
- Pede a confirmação da palavra-passe ao remover a conta ou adicionar novas associações.
- Mostra a hora em que a sugestão foi feita.
- Há uma nova verificação de qualidade para o ponto-e-vírgula à direita.

- Garante que as ligações de pesquisa podem ser compartilhadas.
- Incluídas informações do texto fonte e capturas de ecrã na API.
- Permite sobrescrever traduções por meio de envio pela API.

4.44.11 Weblate 2.13.1

Lançado em 12 de abril de 2017.

- Corrigida a listagem de projetos gerenciados no perfil.
- Corrigido o problema de migração em que algumas permissões estavam faltando.
- Corrigida a listagem do formato de ficheiro atual ao descarregar a tradução.
- Retorna HTTP 404 ao tentar acessar o projeto onde o utilizador não tem privilégios.

4.44.12 Weblate 2.13

Lançado em 12 de abril de 2017.

- Corrigidas verificações de qualidade em modelos de tradução.
- Adicionada verificação de qualidade para acionar na perda de tradução.
- Adiciona uma opção para ver sugestões pendentes de um utilizador.
- Adiciona opção para construir listas de componentes automaticamente.
- O painel padrão para utilizadores não autenticados pode ser configurado.
- Adiciona a opção de navegar por 25 textos aleatórios para revisão.
- O histórico agora indica mudança de texto.
- Melhor relatório de erros ao adicionar uma nova tradução.
- Adicionado pesquisa por idioma dentro do projeto.
- ACLs de grupo agora podem ser limitados a certas permissões.
- Os ACLs por projeto agora são implementados a usar ACL de grupo.
- Adicionado controle de privilégios mais refinado.
- Várias pequenas melhorias na interface do utilizador.

4.44.13 Weblate 2.12

Lançado em 3 de março de 2017.

- Melhorada a interface administrativa para grupos.
- Adicionado suporte à API do Yandex Translate.
- Maior velocidade de pesquisa em todo o site.
- Adicionada pesquisa ampla de projeto e componente.
- Adicionada pesquisa e substituição em todo o projeto e componente.
- Melhorada a renderização de traduções inconsistentes.
- Adicionado suporte para abrir fiheiros fonte no editor local.
- Adicionado suporte para configuração de teclado visual com caracteres especiais.
- Melhorado a gestão de captura de ecrã com suporte OCR para correspondência de textos fonte.

- A mensagem de commit padrão agora inclui informações de tradução e URL.
- Adicionado suporte para o formato de tradução do Joomla.
- Maior confiabilidade de importação em formatos de ficheiros.

4.44.14 Weblate 2.11

Lançado em 31 de janeiro de 2017.

- Inclui informações detalhadas sobre o idioma na página de idioma.
- Melhorias no back-end do Mercurial.
- Adicionada opção para especificar a prioridade do componente de tradução.
- Uso mais consistente da ACL de grupo, mesmo com menos permissões usadas.
- Adicionada variável WL_BRANCH para criar scripts.
- Melhorada a documentação de programador.
- Melhor compatibilidade com várias versões git na extensão do exportador git.
- Incluído por projecto e estatísticas de componentes.
- Adicionado mapeamento de código de idioma para melhor suporte da API do Microsoft Translate.
- A limpeza do texto completo foi movida para o trabalho em segundo plano para tornar a remoção da tradução mais rápida.
- Exibição fixa de fonte plural para idiomas com forma plural única.
- Melhorado o tratamento de erros em import_project.
- Várias melhorias de desempenho.

4.44.15 Weblate 2.10.1

Lançado em 20 de janeiro de 2017.

- Não vaza a existência da conta no formulário de redefinição de palavra-passe (CVE-2017-5537).

4.44.16 Weblate 2.10

Lançado em 15 de dezembro de 2016.

- Adicionada verificação de qualidade para verificar se os plurais são traduzidos de forma diferente.
- Corrigidos ganchos so GitHub para repositórios com autenticação.
- Adicionado módulo opcional de exportador git.
- Suporte para API Microsoft Cognitive Services Translator.
- Simplificadas as interfaces de utilizador de projeto e componente.
- Adicionada correção automática para remover caracteres de controle.
- Adicionada visão geral por idioma ao projeto.
- Adicionado suporte para exportação de CSV.
- Adicionado download de CSV para estatísticas.
- Adicionada visualização de matriz para uma visão geral rápida de todas as traduções.
- Adicionada API básica para mudanças e textos.

- Adicionado suporte ao servidor Apertium Apy para traduções de máquina.

4.44.17 Weblate 2.9

Lançado em 4 de novembro de 2016.

- Parâmetros estendidos para o comando de gestão createadmin.
- Estendido import_json para ser capaz de lidar com componentes existentes.
- Adicionado suporte para ficheiros de YAML.
- Donos de projeto agora podem configurar detalhes de componente e projeto de tradução.
- Usa projetos «Observados» em vez de «Assinados».
- Projetos podem ser observados diretamente a partir da página do projeto.
- Adicionado widget de status multi-idioma.
- Realça o idioma secundário se não mostrar o fonte.
- Regista apagar a sugestão no histórico.
- Melhorada a UX de seleção de idiomas no perfil.
- Corrigida a exibição de mensagens do quadro de comunicações de componente.
- Mantém a guia de preferências selecionada após gravar.
- Mostra o comentário do texto fonte com mais destaque.
- Instala automaticamente o driver de mesclagem Gettext PO para repositórios Git.
- Adicionado recurso de pesquisa e substituição.
- Adicionado suporte para enviar contexto visual (captura de telas) para traduções.

4.44.18 Weblate 2.8

Lançado em 31 de agosto de 2016.

- Melhorias na documentação.
- Traduções.
- Atualizadas as bibliotecas JavaScript empacotadas.
- Adicionado o comando de gestão list_translators.
- Django 1.8 não é mais compatível.
- Corrigida a compatibilidade com Django 1.10.
- Adicionado suporte a Subversion.
- A verificação de validade de XML foi separada de tags XML incompatíveis.
- Corrigida a API para respeitar as configurações de HIDE_REPO_CREDENTIALS.
- Mostra a alteração do fonte no modo Zen.
- Alt+PageUp/PageDown/Home/End agora funciona no modo Zen também.
- Adiciona uma dica de ferramenta mostrando a hora exata das alterações.
- Adiciona uma opção para selecionar filtros e pesquisar na página de tradução.
- Adicionada UI para remoção de tradução.
- Melhorado o comportamento ao inserir objetos colocáveis.

- Corrigidos problemas de bloqueio automático no modo Zen.

4.44.19 Weblate 2.7

Lançado em 10 de julho de 2016.

- Removida a tradução automática da web do Google.
- Melhorada a mensagem de commit ao adicionar tradução.
- Corrigida a API do Google Tradutor para o idioma hebraico.
- Compatibilidade com Mercurial 3.8.
- Adicionado comando de gestão import_json.
- Ordenação correta das traduções listadas.
- Mostra o texto completo da sugestão, não apenas uma diferença.
- Estende a API (status de repositório detalhado, estatísticas, …).
- O conjunto de testes não requer mais acesso à rede para testar repositórios.

4.44.20 Weblate 2.6

Lançado em 28 de abril de 2016.

- Coeeigida a validação de componentes com filtro de idioma.
- Melhorado suporte para ficheiros de XLIFF.
- Corrigida a tradução de máquina para fontes não inglesas.
- Adicionada API REST.
- Compatibilidade com Django 1.10.
- Adicionadas categorias às mensagens do quadro de comunicações.

4.44.21 Weblate 2.5

Lançado em 10 de março de 2016.

- Corrigida tradução de máquina para donos de projetos.
- Melhorado o desempenho de operações de commit e push.
- Novo comando de gestão para adicionar sugestões a partir da linha de comando.
- Adicionado suporte para mesclar comentários ao enviar ficheiros.
- Adicionado suporte para algumas extensões GNU para o formato C printf.
- Melhorias na documentação.
- Adicionado suporte para gerar créditos de tradutores.
- Adicionado suporte para gerar estatísticas de contribuidores.
- A pesquisa em todo o site pode pesquisar apenas num idioma.
- Melhora as verificações de qualidade para armênio.
- Suporte para iniciar componentes de tradução sem traduções existentes.
- Suporte para adicionar novas traduções em Qt TS.
- Melhorado suporte para tradução de ficheiros de PHP.

- Melhorias de desempenho para verificações de qualidade.
- Pesquisa corrigida para todo o site por verificações com falha.
- Adicionada opção para especificar o idioma fonte.
- Melhorado suporte para ficheiros de XLIFF.
- Estendida a lista de opções para import_project.
- Melhoria da segmentação para mensagens do quadro de avisos.
- Suporte para tradução automática entre projetos.
- Otimizado o índice de pesquisa de texto completo.
- Adicionado comando de gestão para a tradução automática.
- Adicionado realce de objetos colocáveis.
- Adicionados atalhos de teclado para os objetos colocáveis, verificações e traduções automáticas.
- Melhorado o bloqueio de tradução.
- Adicionada verificação de qualidade para interpolação AngularJS.
- Adicionadas ACLs extensivas baseadas em grupos .
- Esclarecida a terminologia sobre textis que necessitam edição (anteriormente «fuzzy»).
- Clarified terminology on strings needing action and untranslated strings.
- Suporte para Python 3.
- Removido o suporte ao Django 1.7.
- Removida a dependência do msginit para a criação de novos ficheiros gettext PO.
- Adicionadas visualizações de painel configuráveis.
- Melhoradas notificações sobre erros de análise.
- Adicionada opção para importar componentes com nome duplicado para import_project.
- Melhorado suporte para tradução de ficheiros de PHP.
- Adicionada exportação XLIFF para dicionário.
- Adicionada exportação de XLIFF e gettext PO para todas as traduções.
- Melhorias na documentação.
- Adicionado suporte para atribuições de grupo automáticas configuráveis.
- Melhorada a adição de novas traduções.

4.44.22 Weblate 2.4

Lançado em 20 de setembro de 2015.

- Melhorado o suporte para ficheiros de PHP.
- Capacidade de adicionar ACL a utilizador anônimo.
- Melhorada a capacidade de configuração do comando import_project.
- Adicionado despejo de histórico em CSV.
- Evita erros de copiar/colar com caracteres de espaço em branco.
- Adicionado suporte para webhooks do Bitbucket.
- Controle mais rígido em textos aproximados ao enviar a tradução.
- Várias URLs foram alteradas, deverá ter de atualizar os seus favoritos.

- Os scripts de gancho são executados com a raiz VCS como diretório atual.
- Os scripts de gancho são executados com variáveis de ambiente que descrevem o componente atual.
- Adiciona o comando de gestão para otimizar o índice de texto completo.
- Adicionado suporte para relatórios de erros para Rollbar.
- Os projetos agora podem ter vários donos.
- Donos de projeto podem gerir a si próprio.
- Adicionado suporte para `javascript-format` usado em gettext PO.
- Suporte para adicionar novas traduções em XLIFF.
- Melhorada a autodetecção de formato de ficheiros.
- Estendidos os atalhos de teclado.
- Melhor correspondência de dicionário para vários idiomas.
- Melhorado o layout da maioria das páginas.
- Suporte para adicionar palavras ao dicionário durante a tradução.
- Adicionado suporte para filtragem de idiomas a serem gerenciados pelo Weblate.
- Adicionado suporte para tradução e importação de ficheiros de CSV.
- Reescrito o tratamento de ficheiros estáticos.
- Ligações diretas de login/registo para serviço de terceiros, se for o único.
- Faz commit de alterações pendentes na remoção da conta.
- Adiciona o comando de getão para alterar o nome do site.
- Adiciona a opção para configurar o committer padrão.
- Adiciona um gancho após adicionar uma nova tradução.
- Adiciona opção para especificar vários ficheiros para adicionar para commit.

4.44.23 Weblate 2.3

Lançado em 22 de maio de 2015.

- Removido o suporte a migrações para Django 1.6 e South.
- Suporte para adicionar novas traduções ao usar ficheiros de propriedade Java.
- Permita aceitar sugestão sem edição.
- Melhorado o suporte para Google OAuth 2.0.
- Adicionado suporte para ficheiros .resx da Microsoft.
- O robots.txt padrão ajustado para impedir o grande rastreamento de traduções.
- Fluxo de trabalho simplificado para aceitar sugestões.
- Adicionados donos de projetos que sempre recebem notificações importantes.
- Permite desativar a edição do modelo monolíngue.
- Visualização mais detalhada do status do repositório.
- Ligação direta para editar o modelo ao alterar a tradução.
- Permitir adicionar mais permissões aos donos do projeto.
- Permite mostrar o idioma secundário no modo Zen.
- Suporte para ocultar o texto fonte em favor do idioma secundário.

4.44.24 Weblate 2.2

Lançado em 19 de fevereiro de 2015.

- Melhorias de desempenho.
- Pesquisa de texto completo nos campos de localização e comentários.
- Novos gráficos de atividades baseados em SVG/JavaScript.
- Suporte para Django 1.8.
- Suporte para exclusão de comentários.
- Adicionado o próprio emblema SVG.
- Adicionado suporte para o Google Analytics.
- Melhorado o tratamento de nomes de ficheiros de tradução.
- Adicionado suporte para traduções JSON monolíngues.
- Regista o bloqueio de componentes num histórico.
- Suporte para edição de idioma fonte (modelo) para traduções monolíngues.
- Adicionado suporte básico para Gerrit.

4.44.25 Weblate 2.1

Lançado em 5 de dezembro de 2014.

- Adicionado suporte para repositórios Mercurial.
- Fonte Glyphicon substituída por Awesome.
- Adicionados ícones para serviços de autenticação social.
- Melhor consistência das cores e ícones dos botões.
- Melhorias na documentação.
- Várias correções de bugs.
- Ocultação automática de colunas na listagem de tradução para telas pequenas.
- Alterada a configuração dos caminhos do sistema de ficheiros.
- Melhorado o tratamento e armazenamento de chaves SSH.
- Melhorado o bloqueio de repositório.
- Verificações de qualidade personalizadas por texto fonte.
- Permite ocultar traduções concluídas a partir do painel.

4.44.26 Weblate 2.0

Lançado em 6 de novembro de 2014.

- Nova UI responsiva usando Bootstrap.
- Backend VCS reescrito.
- Melhorias na documentação.
- Adicionado quadro de comunicações para mensagens para todo o site.
- Prioridade de textos configurável.
- Adicionado suporte ao formato de ficheiros JSON.

- Corrigida a geração de ficheiros mo em certos casos.
- Adicionado suporte a notificações do GitLab.
- Adicionado suporte a desativar sugestões de tradução.
- Suporte ao Django 1.7.
- Os projetos com ACL agora têm gestão de utilizadores.
- Estendidas as possibilidades de pesquisa.
- Fornece mais dicas aos tradutores sobre os plurais.
- Corrigido o bloqueio de repositório Git.
- Compatibilidade com versões mais antigas do Git.
- Melhorado o suporte a ACLs.
- Adicionados botões para aspas por idioma e outros caracteres especiais.
- Suporte para exportação de estatísticas como JSONP.

4.45 Série 1.x do Weblate

4.45.1 Weblate 1.9

Lançado em 6 de maio de 2014.

- Compatibilidade com Django 1.6.
- Compatibilidade descontinuada com Django 1.4.
- Comandos de gestão para bloquear/desbloquear traduções.
- Melhorado suporte para ficheiros de Qt TS.
- Os utilizadores agora podem apagar a própria conta.
- Avatares podem ser desativados.
- Mesclados os atributos de nome e sobrenome.
- Avatares agora são obtidos e armazenados em cache no lado do servidor.
- Adicionado suporte para emblema do shields.io.

4.45.2 Weblate 1.8

Lançado em 7 de novembro de 2013.

- Confira o manual para as instruções de atualização.
- Melhor listagem de resumo de projeto.
- Opções mais visíveis para compartilhamento.
- Mais controle sobre privilégios de utilizadores anônimos.
- Suporte a login usando serviços de terceiros, confira o manual para obter mais detalhes.
- Os utilizadores podem fazer login por e-mail em vez de nome de utilizador.
- Melhorias na documentação.
- Melhorada a revisão de textos fontes.
- Pesquisa em todos os textos.

- Melhor rastreamento de textos fonte.
- Proteção por captcha para registar.

4.45.3 Weblate 1.7

Lançado em 7 de outubro de 2013.

- Confira o manual para as instruções de atualização.
- Suporte para verificar a texto de formatação com chaves do Python.
- Personalização por componente de verificações de qualidade.
- Estatísticas detalhadas por tradução.
- Alterada a forma de vinculação de sugestões, verificações e comentários de textos.
- Utilizadores podem agora adicionar texto à mensagem de commit.
- Suporte a inscrição por novas solicitações de idiomas.
- Suporte a adição de novas traduções.
- Widgets e gráficos são agora renderizados usando Pillow em vez de Pango + Cairo.
- Adiciona widget de emblema de status.
- Removida a verificação inválida de direção do texto.
- As alterações no dicionário estão agora registadas no histórico.
- Melhorias de desempenho para a visualização de tradução.

4.45.4 Weblate 1.6

Lançado em 25 de julho de 2013.

- Melhor tratamento de erros ao registar.
- Navegação de mudanças.
- Corrigida a classificação de sugestões de tradução de máquina.
- Melhorado o suporte para tradução de máquina do MyMemory.
- Adicionado suporte para tradução de máquina de Amagama.
- Várias otimizações sobre páginas usadas com frequência.
- Realça frases pesquisadas em resultados de pesquisa.
- Suporte para correções automáticas enquanto salva a mensagem.
- Rastreamento do histórico de tradução e opção para revertê-lo.
- Adicionado suporte para API do Google Tradutor.
- Suporte adicional para a gestão de chaves de host SSH.
- Várias melhorias de validação de formulários.
- Várias melhorias de verificação de qualidade.
- Melhorias de desempenho para importação.
- Adicionado suporte para votação das sugestões.
- Limpeza da interface administrativa.

4.45.5 Weblate 1.5

Lançado em 16 de abril de 2013.

- Confira o manual para as instruções de atualização.
- Adicionadas páginas públicas de utilizadores.
- Melhor nomenclatura de formas plurais.
- Adicionado suporte para exportação de glossário TBX.
- Adicionado suporte para notificações bitbucket.
- Os gráficos de atividade já estão disponíveis para cada tradução, idioma ou utilizador.
- Opções estendidas de comando administrativo import_project.
- Compatível com Django 1.5.
- Avatares são agora mostrados usando libravatar.
- Adicionada possibilidade de impressão bonita da exportação JSON.
- Várias melhorias de desempenho.
- Indica verificações de falha ou textos aproximados em barras de progresso para projetos ou idiomas também.
- Adicionado suporte para ganchos personalizados pré-commit e fazer commit de ficheiros adicionais.
- Reescrita a pesquisa para melhor desempenho e experiência do utilizador.
- Nova interface para traduções de máquina.
- Adicinado suporte para ficheiros po monolíngues.
- Estende a quantidade de metadados armazenados em cache para melhorar a velocidade de várias pesquisas.
- Agora também mostra contagem de palavras.

4.45.6 Weblate 1.4

Lançado em 23 de janeiro de 2013.

- Corrigida a exclusão de verificações/comentários sobre exclusão de texto.
- Adicionada opção para desativar a propagação automática de traduções.
- Adicionada opção para se inscrever para falhas de mesclagem.
- Importa corretamente em projetos que precisem de carregador ttkit personalizado.
- Adicionados mapas de site para permitir um acesso mais fácil por rastreadores.
- Fornece ligações diretas para textos em e-mails de notificação ou feeds.
- Várias melhorias na interface administrativa.
- Fornece dicas para a configuração da produção na interface administrativa.
- Adicionados widgets e página de engajamento por idioma.
- Melhorado o manuseio de bloqueio de tradução.
- Mostra trechos de código para widgets em mais variantes.
- Indica verificações de falha ou textos aproximados em barras de progresso.
- Mais opções para formatar mensagem de commit.
- Corrigida a manipulação de erros com serviços de tradução de máquina.
- Melhorado o comportamento de bloqueio de tradução automática.

- Suporte para mostrar alterações do texto fonte anterior.
- Adicionado suporte para pesquisa de subtexto.
- Várias melhorias de verificação de qualidade.
- Suporte para ACL por projeto.
- Cobertura de código básica por testes unitários.

4.45.7 Weblate 1.3

Lançado em 16 de novembro de 2012.

- Compatibilidade com backend de banco de dados PostgreSQL.
- Remove idiomas removidos no repositório git upstream.
- Melhoria do processamento dos controlos de qualidade.
- Adicionadas novas verificações (BBCode, marcação XML e novas linhas).
- Suporte para opcionalmente fazer rebase em vez de mesclar.
- Possibilidade de realocar o Weblate (por exemplo, executá-lo no caminho /weblate).
- Suporte para escolher manualmente o tipo de ficheiro no caso de falha na autodetecção.
- Melhor suporte para recursos Android.
- Suporte para gerar chave SSH a partir da interface web.
- Exportações de dados mais visíveis.
- Novos botões para inserir alguns caracteres especiais.
- Suporte para exportação de dicionário.
- Suporte para bloquear toda a instalação do Weblate.
- Verificações para texto fonte e suporte para revisão de texto fonte.
- Suporte para comentários de utilizadores para traduções e textos fonte.
- Melhor rastreamento de logs de alteração.
- As alterações agora podem ser monitoradas a usar RSS.
- Melhorado o suporte para idiomas RTL.

4.45.8 Weblate 1.2

Lançado em 14 de agosto de 2012.

- O Weblate agora usa o South para migração do banco de dados. Confira as instruções de atualização se estiver a atualizar.
- Corrigidos pequenos problemas com repositórios git vinculados.
- Nova página de introdução para engajar as pessoas com tradução a usar Weblate.
- Adicionados widgets que podem ser usados para promover projetos de tradução.
- Adicionada opção para redefinir o repositório para a origem (para utilizadores privilegiados).
- O projeto ou componente agora pode ser bloqueado para traduções.
- Possibilidade de desativar algumas traduções.
- Opções configuráveis para adicionar novas traduções.
- Configuração de git commits por projeto.

- Proteção antispam simples.
- Melhor layout da página principal.
- Suporte para enviar automaticamente mudanças em cada commit.
- Suporte para notificações por e-mail de tradutores.
- Lista apenas os idiomas usados nas preferências.
- Melhorado o tratamento de idiomas desconhecidos durante a importação de projetos.
- Suporte para bloqueio de tradução por tradutor.
- Opcionalmente, mantém o cabeçalho `Language-Team` no ficheiro po.
- Inclui algumas estatísticas na página Sobre.
- Adiciona suporte (e exige) django-registration 0.8.
- Fazer cache de contagens de cadeias a conter verificações com falha.
- Verificação de requisitos durante a configuração.
- Melhorias na documentação.

4.45.9 Weblate 1.1

Lançado em 4 de julho de 2012.

- Melhoradas várias traduções.
- Melhor validação durante a criação do componente.
- Adicionado suporte para repositórios git compartilhados entre componentes.
- Não é necessário fazer commit em todas as tentativas de fazer pull do repositório remoto.
- Adicionado suporte para descarregar a indexação.

4.45.10 Weblate 1.0

Lançado em 10 de maio de 2012.

- Melhirada a validação ao adicionar/gravar componente.
- Suporte experimental para ficheiros de componentes do Android (precisa de ttkit com patch aplicado).
- As atualizações dos ganchos são executadas em segundo plano.
- Melhoradas as instruções de instalação.
- Melhorada a navegação no dicionário.

4.46 Série 0.x do Weblate

4.46.1 Weblate 0.9

Lançado em 18 de abril de 2012.

- Corrigida a importação de idiomas desconhecidos.
- Melhorada a lista de mensagens próximas.
- Melhoradas várias verificações.
- Atualizações de documentação.

- Adicionada definição para vários outros idiomas.
- Várias limpezas de código.
- Melhorias na documentação.
- Alteração do layout do ficheiro.
- Atualiza scripts auxiliares para Django 1.4.
- Melhorada a navegação durante a tradução.
- Melhor tratamento de renomeações de ficheiros po.
- Melhor validação durante a criação do componente.
- Integrada a configuração completa no syncdb.
- Adicionada lista de mudanças recentes para todas as páginas de tradução.
- Check for untranslated strings ignores format string only messages.

4.46.2 Weblate 0.8

Lançado em 3 de abril de 2012.

- Substituída a própria pesquisa de texto completo por Whoosh.
- Várias correções e melhorias nas verificações.
- Novo comando updatechecks.
- Muitas atualizações de tradução.
- Adicionado dicionário para armazenar os termos usados com mais frequência.
- Adicionado /admin/report/ para uma visão geral do status dos repositórios.
- Os serviços de tradução de máquina não bloqueiam mais o carregamento da página.
- A interface de gestão agora também contém ações úteis para atualizar dados.
- Regista o log de alterações feitas pelos utilizadores.
- Capacidade de adiar o commit para Git para gerar menos commits de um único utilizador.
- Possibilidade de navegar nas verificações com falha.
- Tradução automática a usar textos já traduzidos.
- Novo sobre a página que mostra as versões usadas.
- Compatibilidade com Django 1.4.
- Capacidade de enviar alterações ao repositório remoto a partir da interface web.
- Adicionada revisão de traduções feitas por outros.

4.46.3 Weblate 0.7

Lançado em 16 de fevereiro de 2012.

- Suporte direto para notificações GitHub.
- Adicionado suporte para limpar verificações e traduções órfãos.
- Exibe textos próximos durante a tradução.
- Exibe textos semelhantes durante a tradução.
- Melhorada pesquisa de texto.

4.46.4 Weblate 0.6

Lançado em 14 de fevereiro de 2012.

- Adicionadas várias verificações para mensagens traduzidas.
- Controle de acesso ajustável.
- Melhirado o tratamento de traduções com novas linhas.
- Adicionada classificação de tabelas do lado do cliente.
- Verifique as instruções de atualização caso esteja a atualizar.

4.46.5 Weblate 0.5

Lançado em 12 de fevereiro de 2012.

- **Suporte para tradução de máquina a usar os seguintes serviços online:**
  - Apertium
  - Microsoft Translator
  - MyMemory
- Várias traduções novas.
- Melhorada mesclagem de mudanças do upstream.
- Melhor tratamento de git pull e tradução simultâneos.
- A propagação também funciona para mudanças aproximadas.
- A propagação também funciona para envio de ficheiros.
- Descarregas de ficheiros corrigidos ao usar FastCGI (e possivelmente outros).

4.46.6 Weblate 0.4

Lançado em 8 de fevereiro de 2012.

- Adicionado guia de uso à documentação.
- Corrigidos ganchos de API para não exigir proteção CSRF.

4.46.7 Weblate 0.3

Lançado em 8 de fevereiro de 2012.

- Melhor exibição da fonte para traduções plurais.
- Nova documentação em formato Sphinx.
- Exibe idiomas secundários durante a tradução.
- Melhorada a página de erro para fornecer uma lista de projetos existentes.
- Novas estatísticas por idioma.

4.46.8 Weblate 0.2

Lançado em 7 de fevereiro de 2012.

- Melhorada a validação de vários formulários.
- Avisa os utilizadores sobre a atualização do perfil.
- Lembre-se de URL para fazer o login.
- Nomenclatura de áreas de texto ao inserir formas plurais.
- Expansão automática da área de tradução.

4.46.9 Weblate 0.1

Lançado em 6 de fevereiro de 2012.

- Lançamento inicial.

Índice de Módulos do Python

W

`wlc`, 160
`wlc.config`, 161
`wlc.main`, 161

HTTP Routing Table

/

ANY /, 107

/api

GET /api/, 110

/api/addons

GET /api/addons/, 147
GET /api/addons/(int:id)/, 147
PUT /api/addons/(int:id)/, 148
DELETE /api/addons/(int:id)/, 148
PATCH /api/addons/(int:id)/, 148

/api/changes

GET /api/changes/, 144
GET /api/changes/(int:id)/, 144

/api/component-lists

GET /api/component-lists/, 148
GET /api/component-lists/(str:slug)/, 148
POST /api/component-lists/(str:slug)/components/, 149
PUT /api/component-lists/(str:slug)/, 149
DELETE /api/component-lists/(str:slug)/, 149
DELETE /api/component-lists/(str:slug)/components/(str:component_slug), 149
PATCH /api/component-lists/(str:slug)/, 149

/api/components

GET /api/components/, 126
GET /api/components/(string:project)/(string:comp, 126
GET /api/components/(string:project)/(string:comp, 130
GET /api/components/(string:project)/(string:comp, 131
GET /api/components/(string:project)/(string:compo, 135
GET /api/components/(string:project)/(string:compo, 131
GET /api/components/(string:project)/(string:compo, 133
GET /api/components/(string:project)/(string:compo, 133
GET /api/components/(string:project)/(string:compo, 132
GET /api/components/(string:project)/(string:compo, 131
GET /api/components/(string:project)/(string:compo, 135
GET /api/components/(string:project)/(string:compo, 134
POST /api/components/(string:project)/(string:comp, 147
POST /api/components/(string:project)/(string:comp, 136
POST /api/components/(string:project)/(string:comp, 131
POST /api/components/(string:project)/(string:comp, 132
POST /api/components/(string:project)/(string:comp, 134
PUT /api/components/(string:project)/(string:compo, 130
DELETE /api/components/(string:project)/(string:c, 130
DELETE /api/components/(string:project)/(string:c, 136
PATCH /api/components/(string:project)/(string:com, 128

/api/groups

GET /api/groups/, 114
GET /api/groups/(int:id)/, 114
POST /api/groups/, 114
POST /api/groups/(int:id)/componentlists/, 116
POST /api/groups/(int:id)/components/, 116

POST /api/groups/(int:id)/languages/, 116
POST /api/groups/(int:id)/projects/, 116
POST /api/groups/(int:id)/roles/, 115
PUT /api/groups/(int:id)/, 115
DELETE /api/groups/(int:id)/, 115
DELETE /api/groups/(int:id)/componentlists/(int:component_list_id), 117
DELETE /api/groups/(int:id)/components/(int:component_id), 116
DELETE /api/groups/(int:id)/languages/(string:language_code), 116
DELETE /api/groups/(int:id)/projects/(int:project_id), 116
PATCH /api/groups/(int:id)/, 115

/api/languages

GET /api/languages/, 118
GET /api/languages/(string:language)/, 118
GET /api/languages/(string:language)/statistics/, 119
POST /api/languages/, 118
PUT /api/languages/(string:language)/, 119
DELETE /api/languages/(string:language)/, 119
PATCH /api/languages/(string:language)/, 119

/api/memory

GET /api/memory/, 142
DELETE /api/memory/(int:memory_object_id)/, 142

/api/metrics

GET /api/metrics/, 150

/api/projects

GET /api/projects/, 120
GET /api/projects/(string:project)/, 120
GET /api/projects/(string:project)/changes/, 121
GET /api/projects/(string:project)/components/, 123
GET /api/projects/(string:project)/languages/, 125
GET /api/projects/(string:project)/repository/, 122
GET /api/projects/(string:project)/statistics/, 126
POST /api/projects/, 120
POST /api/projects/(string:project)/components/, 123
POST /api/projects/(string:project)/repository/, 122
PUT /api/projects/(string:project)/, 121
DELETE /api/projects/(string:project)/, 121
PATCH /api/projects/(string:project)/, 121

/api/roles

GET /api/roles/, 117
GET /api/roles/(int:id)/, 117
POST /api/roles/, 117
PUT /api/roles/(int:id)/, 117
DELETE /api/roles/(int:id)/, 118
PATCH /api/roles/(int:id)/, 118

/api/screenshots

GET /api/screenshots/, 145
GET /api/screenshots/(int:id)/, 145
GET /api/screenshots/(int:id)/file/, 145
POST /api/screenshots/, 146
POST /api/screenshots/(int:id)/file/, 145
POST /api/screenshots/(int:id)/units/, 146
PUT /api/screenshots/(int:id)/, 147
DELETE /api/screenshots/(int:id)/, 147
DELETE /api/screenshots/(int:id)/units/(int:unit_
146
PATCH /api/screenshots/(int:id)/, 146

/api/tasks

GET /api/tasks/, 150
GET /api/tasks/(str:uuid)/, 150

/api/translations

GET /api/translations/, 136
GET /api/translations/(string:project)/(string:co
136
GET /api/translations/(string:project)/(string:co
139
GET /api/translations/(string:project)/(string:co
140
GET /api/translations/(string:project)/(string:co
141
GET /api/translations/(string:project)/(string:co
141
GET /api/translations/(string:project)/(string:co
139
POST /api/translations/(string:project)/(string:c
139
POST /api/translations/(string:project)/(string:c
140
POST /api/translations/(string:project)/(string:c
141
POST /api/translations/(string:project)/(string:c
139

DELETE /api/translations/(string:project)/(string:component)/(string:language)/, 138
POST /hooks/gitlab/, 151
POST /hooks/pagure/, 152

/api/units

GET /api/units/, 142
GET /api/units/(int:id)/, 142
PUT /api/units/(int:id)/, 144
DELETE /api/units/(int:id)/, 144
PATCH /api/units/(int:id)/, 143

/api/users

GET /api/users/, 110
GET /api/users/(str:username)/, 111
GET /api/users/(str:username)/notifications/, 113
GET /api/users/(str:username)/notifications/(int:subscription_id)/, 113
GET /api/users/(str:username)/statistics/, 112
POST /api/users/, 110
POST /api/users/(str:username)/groups/, 112
POST /api/users/(str:username)/notifications/, 113
PUT /api/users/(str:username)/, 111
PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 113
DELETE /api/users/(str:username)/, 112
DELETE /api/users/(str:username)/groups/, 112
DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 113
PATCH /api/users/(str:username)/, 112
PATCH /api/users/(str:username)/notifications/(int:subscription_id)/, 113

/exports

GET /exports/rss/, 154
GET /exports/rss/(string:project)/, 154
GET /exports/rss/(string:project)/(string:component)/, 154
GET /exports/rss/(string:project)/(string:component)/(string:language)/, 154
GET /exports/rss/language/(string:language)/, 154
GET /exports/stats/(string:project)/(string:component)/, 153

/hooks

GET /hooks/update/(string:project)/, 151
GET /hooks/update/(string:project)/(string:component)/, 151
POST /hooks/azure/, 152
POST /hooks/bitbucket/, 151
POST /hooks/gitea/, 152
POST /hooks/gitee/, 152
POST /hooks/github/, 151

Índice

Símbolos

.XML resource file
    file format, 92
--add
    auto_translate command line option, 384
--addon
    install_addon command line option, 391
--age
    commit_pending command line option, 385
--author
    add_suggestions command line option, 384
--author-email
    wlc command line option, 158
--author-name
    wlc command line option, 158
--base-file-template
    import_project command line option, 389
--check
    importusers command line option, 391
--config
    wlc command line option, 156
--config-section
    wlc command line option, 156
--configuration
    install_addon command line option, 391
--convert
    wlc command line option, 157
--email
    createadmin command line option, 386
--file-format
    import_project command line option, 389
--force
    loadpo command line option, 392
--force-commit
    pushgit command line option, 393
--format
    wlc command line option, 156
--fuzzy
    wlc command line option, 158
--ignore
    import_json command line option, 387
--inconsistent
    auto_translate command line option, 384
--input
    wlc command line option, 158
--key
    wlc command line option, 156
--lang
    loadpo command line option, 392
--language-code
    list_translators command line option, 392
--language-map
    import_memory command line option, 388
--language-regex
    import_project command line option, 389
--license
    import_project command line option, 389
--license-url
    import_project command line option, 389
--main-component
    import_json command line option, 387
    import_project command line option, 389
--method
    wlc command line option, 158
--mode
    auto_translate command line option, 385
--mt
    auto_translate command line option, 385
--name
    createadmin command line option, 386
--name-template

import_project command line option, 389
--new-base-template
import_project command line option, 389
--no-password
createadmin command line option, 386
--no-privs-update
setupgroups command line option, 394
--no-projects-update
setupgroups command line option, 394
--no-update
setuplang command line option, 394
--output
wlc command line option, 157
--overwrite
auto_translate command line option, 384
wlc command line option, 158
--password
createadmin command line option, 386
--project
import_json command line option, 387
--source
auto_translate command line option, 384
--threshold
auto_translate command line option, 385
--update
createadmin command line option, 386
import_json command line option, 387
install_addon command line option, 391
--url
wlc command line option, 156
--user
auto_translate command line option, 384
--username
createadmin command line option, 386
--vcs
import_project command line option, 389

A

add_suggestions
weblate admin command, 384
add_suggestions command line option
--author, 384
ADMINS
setting, 209
AKISMET_API_KEY
setting, 342
ALLOWED_HOSTS
setting, 209
Android
file format, 87
ANONYMOUS_USER_NAME
setting, 342
API, 107, 155, 160
Apple strings
file format, 87
ARB
file format, 91
AUDITLOG_EXPIRY
setting, 343
AUTH_LOCK_ATTEMPTS
setting, 343
AUTH_TOKEN_VALID
setting, 344
auto_translate
weblate admin command, 384
auto_translate command line option
--add, 384
--inconsistent, 384
--mode, 385
--mt, 385
--overwrite, 384
--source, 384
--threshold, 385
--user, 384
AUTO_UPDATE
setting, 343
AUTOFIX_LIST
setting, 344
AVATAR_URL_PREFIX
setting, 343

B

BACKGROUND_TASKS
setting, 345
BaseAddon (*classe em weblate.addons.base*), 429
BASIC_LANGUAGES
setting, 345
bilingual
translation, 78
BORG_EXTRA_ARGS
setting, 345

C

can_install() (*método de classe weblate.addons.base.BaseAddon*), 429
CELERY_BACKUP_OPTIONS, 167, 183
CELERY_BEAT_OPTIONS, 167, 183
CELERY_MAIN_OPTIONS, 167, 183
CELERY_MEMORY_OPTIONS, 167, 183
CELERY_NOTIFY_OPTIONS, 167, 183
celery_queues
weblate admin command, 385
CELERY_TRANSLATE_OPTIONS, 167, 183
changes
wlc command line option, 157
CHECK_LIST
setting, 346
checkgit

weblate admin command, 385
cleanup
wlc command line option, 157
cleanup_ssh_keys
weblate admin command, 386
cleanuptrans
weblate admin command, 386
Comma separated values
file format, 92
Command (*classe em wlc.main*), 161
COMMENT_CLEANUP_DAYS
setting, 347
commit
wlc command line option, 157
commit_pending
weblate admin command, 385
commit_pending command line option
--age, 385
COMMIT_PENDING_HOURS
setting, 347
commitgit
weblate admin command, 385
configure() (*método weblate.addons.base.BaseAddon*), 429
CONTACT_FORM
setting, 347
createadmin
weblate admin command, 386
createadmin command line option
--email, 386
--name, 386
--no-password, 386
--password, 386
--update, 386
--username, 386
CSP_CONNECT_SRC
setting, 346
CSP_FONT_SRC
setting, 346
CSP_IMG_SRC
setting, 346
CSP_SCRIPT_SRC
setting, 346
CSP_STYLE_SRC
setting, 346
CSV
file format, 92

D

daily() (*método weblate.addons.base.BaseAddon*), 429
DATA_DIR
setting, 347
DATABASE_BACKUP
setting, 348
DATABASES
setting, 210
DEBUG
setting, 210
DEFAULT_ACCESS_CONTROL
setting, 348
DEFAULT_ADD_MESSAGE
setting, 349
DEFAULT_ADDON_MESSAGE
setting, 349
DEFAULT_ADDONS
setting, 349
DEFAULT_AUTO_WATCH
setting, 348
DEFAULT_COMMIT_MESSAGE
setting, 349
DEFAULT_COMMITER_EMAIL
setting, 349
DEFAULT_COMMITER_NAME
setting, 350
DEFAULT_DELETE_MESSAGE
setting, 349
DEFAULT_FROM_EMAIL
setting, 210
DEFAULT_LANGUAGE
setting, 350
DEFAULT_MERGE_MESSAGE
setting, 349
DEFAULT_MERGE_STYLE
setting, 350
DEFAULT_PAGE_LIMIT
setting, 358
DEFAULT_PULL_MESSAGE
setting, 351
DEFAULT_RESTRICTED_COMPONENT
setting, 349
DEFAULT_SHARED_TM
setting, 350
DEFAULT_TRANSLATION_PROPAGATION
setting, 350
download
wlc command line option, 157
DTD
file format, 94
dump_memory
weblate admin command, 387
dumpuserdata
weblate admin command, 387

E

ENABLE_AVATARS
setting, 351
ENABLE_HOOKS
setting, 351
ENABLE_HTTPS
setting, 351
ENABLE_SHARING
setting, 351

F

file format

.XML resource file, 92
Android, 87
Apple strings, 87
ARB, 91
Comma separated values, 92
CSV, 92
DTD, 94
gettext, 81
go-i18n, 90
GWT properties, 85
i18next, 90
INI translations, 85
Java properties, 84
Joomla translations, 86
JSON, 88
mi18n lang, 84
PHP strings, 88
PO, 81
Qt, 86
RC, 95
ResourceDictionary, 92
RESX, 92
Ruby YAML, 94
Ruby YAML Ain't Markup Language, 94
string resources, 87
TS, 86
WPF, 92
XLIFF, 82
XML, 95
YAML, 93
YAML Ain't Markup Language, 93

G

get() (*método wlc.Weblate*), 160
get_add_form() (*método de classe weblate.addons.base.BaseAddon*), 429
GET_HELP_URL
setting, 352
get_settings_form() (*método weblate.addons.base.BaseAddon*), 429
gettext
file format, 81
GITEA_CREDENTIALS
setting, 352
GITEA_TOKEN
setting, 352
GITEA_USERNAME
setting, 352
GITHUB_CREDENTIALS
setting, 353
GITHUB_TOKEN
setting, 354
GITHUB_USERNAME
setting, 353
GITLAB_CREDENTIALS
setting, 352
GITLAB_TOKEN
setting, 353
GITLAB_USERNAME
setting, 353
go-i18n
file format, 90
GOOGLE_ANALYTICS_ID
setting, 354
GWT properties
file format, 85

H

HIDE_REPO_CREDENTIALS
setting, 354
HIDE_VERSION
setting, 354

I

i18next
file format, 90
import_demo
weblate admin command, 387
import_json
weblate admin command, 387
import_json command line option
--ignore, 387
--main-component, 387
--project, 387
--update, 387
import_memory
weblate admin command, 388
import_memory command line option
--language-map, 388
import_project
weblate admin command, 388
import_project command line option
--base-file-template, 389
--file-format, 389
--language-regex, 389
--license, 389
--license-url, 389
--main-component, 389
--name-template, 389
--new-base-template, 389
--vcs, 389
importuserdata
weblate admin command, 391
importusers
weblate admin command, 391
importusers command line option
--check, 391
INI translations
file format, 85
install_addon
weblate admin command, 391
install_addon command line option
--addon, 391
--configuration, 391
--update, 391
INTERLEDGER_PAYMENT_POINTERS

setting, 354
iOS
translation, 87
IP_BEHIND_REVERSE_PROXY
setting, 355
IP_PROXY_HEADER
setting, 355
IP_PROXY_OFFSET
setting, 355

J

Java properties
file format, 84
Joomla translations
file format, 86
JSON
file format, 88

L

LEGAL_URL
setting, 356
LICENSE_EXTRA
setting, 356
LICENSE_FILTER
setting, 356
LICENSE_REQUIRED
setting, 357
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
setting, 357
list_languages
weblate admin command, 391
list_translators
weblate admin command, 392
list_translators command line option
--language-code, 392
list_versions
weblate admin command, 392
list-components
wlc command line option, 156
list-languages
wlc command line option, 156
list-projects
wlc command line option, 156
list-translations
wlc command line option, 156
load() (*método wlc.config.WeblateConfig*), 161
loadpo
weblate admin command, 392
loadpo command line option
--force, 392
--lang, 392
LOCALIZE_CDN_PATH
setting, 357
LOCALIZE_CDN_URL
setting, 357
lock
wlc command line option, 157
lock_translation
weblate admin command, 392
lock-status
wlc command line option, 157
LOGIN_REQUIRED_URLS
setting, 357
LOGIN_REQUIRED_URLS_EXCEPTIONS
setting, 357
ls
wlc command line option, 157

M

main() (*no módulo wlc.main*), 161
MATOMO_SITE_ID
setting, 358
MATOMO_URL
setting, 358
mi18n lang
file format, 84
monolingual
translation, 78
move_language
weblate admin command, 393
módulo
wlc, 160
wlc.config, 161
wlc.main, 161

N

NEARBY_MESSAGES
setting, 358

P

PAGURE_CREDENTIALS
setting, 359
PAGURE_TOKEN
setting, 359
PAGURE_USERNAME
setting, 359
PHP strings
file format, 88
PIWIK_SITE_ID
setting, 358
PIWIK_URL
setting, 358
PO
file format, 81
post() (*método wlc.Weblate*), 160
post_add() (*método weblate.addons.base.BaseAddon*), 429
post_commit() (*método weblate.addons.base.BaseAddon*), 429
post_push() (*método weblate.addons.base.BaseAddon*), 429
post_update() (*método weblate.addons.base.BaseAddon*), 429
pre_commit() (*método weblate.addons.base.BaseAddon*), 430

pre_push() (*método weblate.addons.base.BaseAddon*), 430
pre_update() (*método weblate.addons.base.BaseAddon*), 430
PRIVACY_URL
    setting, 359
PROJECT_BACKUP_KEEP_COUNT
    setting, 360
PROJECT_BACKUP_KEEP_DAYS
    setting, 360
pull
    wlc command line option, 157
push
    wlc command line option, 157
pushgit
    weblate admin command, 393
pushgit command line option
    --force-commit, 393
Python, 160
Python Enhancement Proposals
    PEP 484, 426, 440

Q

Qt
    file format, 86

R

RATELIMIT_ATTEMPTS
    setting, 360
RATELIMIT_LOCKOUT
    setting, 360
RATELIMIT_WINDOW
    setting, 360
RC
    file format, 95
REDIS_PASSWORD, 180
register_command() (*no módulo wlc.main*), 161
REGISTRATION_ALLOW_BACKENDS
    setting, 360
REGISTRATION_CAPTCHA
    setting, 361
REGISTRATION_EMAIL_MATCH
    setting, 361
REGISTRATION_OPEN
    setting, 361
repo
    wlc command line option, 157
REPOSITORY_ALERT_THRESHOLD
    setting, 362
REQUIRE_LOGIN
    setting, 362
reset
    wlc command line option, 157
ResourceDictionary
    file format, 92
REST, 107
RESX
    file format, 92
RFC
    RFC 5646, 77
Ruby YAML
    file format, 94
Ruby YAML Ain't Markup Language
    file format, 94

S

save_state() (*método weblate.addons.base.BaseAddon*), 430
SECRET_KEY
    setting, 210
SENTRY_DSN
    setting, 362
SERVER_EMAIL
    setting, 210
SESSION_COOKIE_AGE_AUTHENTICATED
    setting, 362
SESSION_ENGINE
    setting, 210
setting
    ADMINS, 209
    AKISMET_API_KEY, 342
    ALLOWED_HOSTS, 209
    ANONYMOUS_USER_NAME, 342
    AUDITLOG_EXPIRY, 343
    AUTH_LOCK_ATTEMPTS, 343
    AUTH_TOKEN_VALID, 344
    AUTO_UPDATE, 343
    AUTOFIX_LIST, 344
    AVATAR_URL_PREFIX, 343
    BACKGROUND_TASKS, 345
    BASIC_LANGUAGES, 345
    BORG_EXTRA_ARGS, 345
    CHECK_LIST, 346
    COMMENT_CLEANUP_DAYS, 347
    COMMIT_PENDING_HOURS, 347
    CONTACT_FORM, 347
    CSP_CONNECT_SRC, 346
    CSP_FONT_SRC, 346
    CSP_IMG_SRC, 346
    CSP_SCRIPT_SRC, 346
    CSP_STYLE_SRC, 346
    DATA_DIR, 347
    DATABASE_BACKUP, 348
    DATABASES, 210
    DEBUG, 210
    DEFAULT_ACCESS_CONTROL, 348
    DEFAULT_ADD_MESSAGE, 349
    DEFAULT_ADDON_MESSAGE, 349
    DEFAULT_ADDONS, 349
    DEFAULT_AUTO_WATCH, 348
    DEFAULT_COMMIT_MESSAGE, 349
    DEFAULT_COMMITER_EMAIL, 349
    DEFAULT_COMMITER_NAME, 350
    DEFAULT_DELETE_MESSAGE, 349
    DEFAULT_FROM_EMAIL, 210
    DEFAULT_LANGUAGE, 350

DEFAULT_MERGE_MESSAGE, 349
DEFAULT_MERGE_STYLE, 350
DEFAULT_PAGE_LIMIT, 358
DEFAULT_PULL_MESSAGE, 351
DEFAULT_RESTRICTED_COMPONENT, 349
DEFAULT_SHARED_TM, 350
DEFAULT_TRANSLATION_PROPAGATION, 350
ENABLE_AVATARS, 351
ENABLE_HOOKS, 351
ENABLE_HTTPS, 351
ENABLE_SHARING, 351
GET_HELP_URL, 352
GITEA_CREDENTIALS, 352
GITEA_TOKEN, 352
GITEA_USERNAME, 352
GITHUB_CREDENTIALS, 353
GITHUB_TOKEN, 354
GITHUB_USERNAME, 353
GITLAB_CREDENTIALS, 352
GITLAB_TOKEN, 353
GITLAB_USERNAME, 353
GOOGLE_ANALYTICS_ID, 354
HIDE_REPO_CREDENTIALS, 354
HIDE_VERSION, 354
INTERLEDGER_PAYMENT_POINTERS, 354
IP_BEHIND_REVERSE_PROXY, 355
IP_PROXY_HEADER, 355
IP_PROXY_OFFSET, 355
LEGAL_URL, 356
LICENSE_EXTRA, 356
LICENSE_FILTER, 356
LICENSE_REQUIRED, 357
LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 357
LOCALIZE_CDN_PATH, 357
LOCALIZE_CDN_URL, 357
LOGIN_REQUIRED_URLS, 357
LOGIN_REQUIRED_URLS_EXCEPTIONS, 357
MATOMO_SITE_ID, 358
MATOMO_URL, 358
NEARBY_MESSAGES, 358
PAGURE_CREDENTIALS, 359
PAGURE_TOKEN, 359
PAGURE_USERNAME, 359
PIWIK_SITE_ID, 358
PIWIK_URL, 358
PRIVACY_URL, 359
PROJECT_BACKUP_KEEP_COUNT, 360
PROJECT_BACKUP_KEEP_DAYS, 360
RATELIMIT_ATTEMPTS, 360
RATELIMIT_LOCKOUT, 360
RATELIMIT_WINDOW, 360
REGISTRATION_ALLOW_BACKENDS, 360
REGISTRATION_CAPTCHA, 361
REGISTRATION_EMAIL_MATCH, 361
REGISTRATION_OPEN, 361
REPOSITORY_ALERT_THRESHOLD, 362
REQUIRE_LOGIN, 362
SECRET_KEY, 210
SENTRY_DSN, 362
SERVER_EMAIL, 210
SESSION_COOKIE_AGE_AUTHENTICATED, 362
SESSION_ENGINE, 210
SIMPLIFY_LANGUAGES, 362
SINGLE_PROJECT, 363
SITE_DOMAIN, 362
SITE_TITLE, 363
SPECIAL_CHARS, 363
SSH_EXTRA_ARGS, 363
STATUS_URL, 364
SUGGESTION_CLEANUP_DAYS, 364
UPDATE_LANGUAGES, 364
URL_PREFIX, 364
VCS_BACKENDS, 365
VCS_CLONE_DEPTH, 365
WEBLATE_ADDONS, 365
WEBLATE_EXPORTERS, 366
WEBLATE_FORMATS, 366
WEBLATE_GPG_IDENTITY, 367
WEBLATE_MACHINERY, 367
WEBSITE_REQUIRED, 367
setupgroups
weblate admin command, 394
setupgroups command line option
--no-privs-update, 394
--no-projects-update, 394
setuplang
weblate admin command, 394
setuplang command line option
--no-update, 394
show
wlc command line option, 156
SIMPLIFY_LANGUAGES
setting, 362
SINGLE_PROJECT
setting, 363
SITE_DOMAIN
setting, 362
SITE_TITLE
setting, 363
SPECIAL_CHARS
setting, 363
SSH_EXTRA_ARGS
setting, 363
stats
wlc command line option, 157
STATUS_URL
setting, 364
store_post_load() (*método weblate.addons.base.BaseAddon*), 430
string resources
file format, 87
SUGGESTION_CLEANUP_DAYS
setting, 364

T

translation
    bilingual, 78
    iOS, 87
    monolingual, 78
TS
    file format, 86

U

unit_pre_create() (*método weblate.addons.base.BaseAddon*), 430
unlock
    wlc command line option, 157
unlock_translation
    weblate admin command, 393
UPDATE_LANGUAGES
    setting, 364
updatechecks
    weblate admin command, 394
updategit
    weblate admin command, 394
upload
    wlc command line option, 157
URL_PREFIX
    setting, 364

V

variável de ambiente
    CELERY_BACKUP_OPTIONS, 167, 183
    CELERY_BEAT_OPTIONS, 167, 183
    CELERY_MAIN_OPTIONS, 167, 183
    CELERY_MEMORY_OPTIONS, 167, 183
    CELERY_NOTIFY_OPTIONS, 167, 183
    CELERY_TRANSLATE_OPTIONS, 167, 183
    POSTGRES_ALTER_ROLE, 179
    POSTGRES_CONN_MAX_AGE, 179
    POSTGRES_DATABASE, 178
    POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 179
    POSTGRES_HOST, 178
    POSTGRES_PASSWORD, 178
    POSTGRES_PASSWORD_FILE, 178
    POSTGRES_PORT, 178
    POSTGRES_SSL_MODE, 178
    POSTGRES_USER, 178
    REDIS_DB, 180
    REDIS_HOST, 179
    REDIS_PASSWORD, 180
    REDIS_PASSWORD_FILE, 180
    REDIS_PORT, 179
    REDIS_TLS, 180
    REDIS_VERIFY_SSL, 180
    ROLLBAR_ENVIRONMENT, 182
    ROLLBAR_KEY, 182
    SENTRY_DSN, 182
    SENTRY_ENVIRONMENT, 182
    SOCIAL_AUTH_SLACK_SECRET, 177
    WEB_WORKERS, 167, 183, 184
    WEBLATE_ADD_ADDONS, 183
    WEBLATE_ADD_APPS, 183
    WEBLATE_ADD_AUTOFIX, 183
    WEBLATE_ADD_CHECK, 183
    WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS, 171
    WEBLATE_ADMIN_EMAIL, 167--169
    WEBLATE_ADMIN_NAME, 167--169
    WEBLATE_ADMIN_PASSWORD, 163, 167--169
    WEBLATE_ADMIN_PASSWORD_FILE, 169
    WEBLATE_AKISMET_API_KEY, 172, 401
    WEBLATE_ALLOWED_HOSTS, 169, 210, 214, 363
    WEBLATE_API_RATELIMIT_ANON, 109, 110, 173
    WEBLATE_API_RATELIMIT_USER, 109, 110, 173
    WEBLATE_AUTH_LDAP_BIND_DN, 174
    WEBLATE_AUTH_LDAP_BIND_PASSWORD, 174
    WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE, 174
    WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERRALS, 174
    WEBLATE_AUTH_LDAP_SERVER_URI, 174
    WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 174
    WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 174
    WEBLATE_AUTH_LDAP_USER_SEARCH, 174
    WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 174
    WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 174
    WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 174
    WEBLATE_AUTO_UPDATE, 181
    WEBLATE_BASIC_LANGUAGES, 173
    WEBLATE_BORG_EXTRA_ARGS, 174
    WEBLATE_CONTACT_FORM, 169
    WEBLATE_CSP_CONNECT_SRC, 172
    WEBLATE_CSP_FONT_SRC, 173
    WEBLATE_CSP_IMG_SRC, 172
    WEBLATE_CSP_SCRIPT_SRC, 172
    WEBLATE_CSP_STYLE_SRC, 173
    WEBLATE_DATABASE_BACKUP, 179
    WEBLATE_DEBUG, 168
    WEBLATE_DEFAULT_ACCESS_CONTROL, 172
    WEBLATE_DEFAULT_AUTO_WATCH, 173
    WEBLATE_DEFAULT_COMMITER_EMAIL, 172
    WEBLATE_DEFAULT_COMMITER_NAME, 172
    WEBLATE_DEFAULT_FROM_EMAIL, 169
    WEBLATE_DEFAULT_PULL_MESSAGE, 172
    WEBLATE_DEFAULT_RESTRICTED_COMPONENT, 172
    WEBLATE_DEFAULT_SHARED_TM, 172
    WEBLATE_DEFAULT_TRANSLATION_PROPAGATION, 172

WEBLATE_EMAIL_BACKEND, 181
WEBLATE_EMAIL_HOST, 180
WEBLATE_EMAIL_HOST_PASSWORD, 181
WEBLATE_EMAIL_HOST_PASSWORD_FILE, 181
WEBLATE_EMAIL_HOST_USER, 180
WEBLATE_EMAIL_PORT, 180, 181
WEBLATE_EMAIL_USE_SSL, 180, 181
WEBLATE_EMAIL_USE_TLS, 180, 181
WEBLATE_ENABLE_AVATARS, 173
WEBLATE_ENABLE_HOOKS, 173
WEBLATE_ENABLE_HTTPS, 170, 243
WEBLATE_ENABLE_SHARING, 174
WEBLATE_GET_HELP_URL, 182
WEBLATE_GITHUB_TOKEN, 171
WEBLATE_GITHUB_USERNAME, 171
WEBLATE_GITLAB_TOKEN, 171
WEBLATE_GITLAB_USERNAME, 171
WEBLATE_GOOGLE_ANALYTICS_ID, 171
WEBLATE_GPG_IDENTITY, 172
WEBLATE_HIDE_VERSION, 173
WEBLATE_INTERLEDGER_PAYMENT_POINTERS, 170
WEBLATE_IP_PROXY_HEADER, 170
WEBLATE_LEGAL_URL, 182
WEBLATE_LICENSE_FILTER, 173
WEBLATE_LICENSE_REQUIRED, 173
WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE, 174
WEBLATE_LOCALIZE_CDN_PATH, 182
WEBLATE_LOCALIZE_CDN_URL, 182
WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 171
WEBLATE_LOGLEVEL, 168
WEBLATE_LOGLEVEL_DATABASE, 168
WEBLATE_NO_EMAIL_AUTH, 178
WEBLATE_PAGURE_TOKEN, 172
WEBLATE_PAGURE_USERNAME, 171
WEBLATE_PRIVACY_URL, 182
WEBLATE_RATELIMIT_ATTEMPTS, 173, 403
WEBLATE_RATELIMIT_LOCKOUT, 173
WEBLATE_RATELIMIT_WINDOW, 173
WEBLATE_REGISTRATION_ALLOW_BACKENDS, 169
WEBLATE_REGISTRATION_OPEN, 169
WEBLATE_REMOVE_ADDONS, 183
WEBLATE_REMOVE_APPS, 183
WEBLATE_REMOVE_AUTOFIX, 183
WEBLATE_REMOVE_CHECK, 183
WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 171
WEBLATE_REQUIRE_LOGIN, 171, 362
WEBLATE_SAML_IDP_ENTITY_ID, 178
WEBLATE_SAML_IDP_IMAGE, 178
WEBLATE_SAML_IDP_TITLE, 178
WEBLATE_SAML_IDP_URL, 178
WEBLATE_SAML_IDP_X509CERT, 178
WEBLATE_SECURE_PROXY_SSL_HEADER, 170
WEBLATE_SERVER_EMAIL, 169
WEBLATE_SERVICE, 167, 184
WEBLATE_SILENCED_SYSTEM_CHECKS, 172, 240
WEBLATE_SIMPLIFY_LANGUAGES, 172
WEBLATE_SITE_DOMAIN, 168, 212, 230, 363
WEBLATE_SITE_TITLE, 168
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY, 176
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET, 176
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY, 177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECR 177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENA 177
WEBLATE_SOCIAL_AUTH_BITBUCKET_KEY, 176
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY, 176
WEBLATE_SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET, 176
WEBLATE_SOCIAL_AUTH_BITBUCKET_SECRET, 176
WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY, 176
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET, 176
WEBLATE_SOCIAL_AUTH_FEDORA, 177
WEBLATE_SOCIAL_AUTH_GITHUB_KEY, 175
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY, 175
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME, 175
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET, 175
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET, 175
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID, 175
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY, 175
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET, 175
WEBLATE_SOCIAL_AUTH_GITLAB_API_URL, 176
WEBLATE_SOCIAL_AUTH_GITLAB_KEY, 176
WEBLATE_SOCIAL_AUTH_GITLAB_SECRET, 176
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY, 176
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET, 176
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_ 176

WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS, 176
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL, 177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM, 177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL, 177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE, 177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY, 177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY, 177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET, 177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE, 177
WEBLATE_SOCIAL_AUTH_OIDC_KEY, 177
WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT, 177
WEBLATE_SOCIAL_AUTH_OIDC_SECRET, 177
WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY, 177
WEBLATE_SOCIAL_AUTH_OPENSUSE, 177
WEBLATE_SOCIAL_AUTH_SLACK_KEY, 177
WEBLATE_SOCIAL_AUTH_UBUNTU, 177
WEBLATE_SSH_EXTRA_ARGS, 174
WEBLATE_STATUS_URL, 182
WEBLATE_TIME_ZONE, 169
WEBLATE_URL_PREFIX, 172
WEBLATE_WEBSITE_REQUIRED, 173
WEBLATE_WORKERS, 167, 183, 184
WL_BRANCH, 339
WL_COMPONENT_NAME, 339
WL_COMPONENT_SLUG, 339
WL_COMPONENT_URL, 339
WL_ENGAGE_URL, 339
WL_FILE_FORMAT, 339
WL_FILEMASK, 339
WL_LANGUAGE, 339
WL_NEW_BASE, 339
WL_PATH, 339
WL_PREVIOUS_HEAD, 339
WL_PROJECT_NAME, 339
WL_PROJECT_SLUG, 339
WL_REPO, 339
WL_TEMPLATE, 339
WL_VCS, 339
VCS_BACKENDS
setting, 365
VCS_CLONE_DEPTH
setting, 365
version
wlc command line option, 156

W

WEB_WORKERS, 167, 183
weblate (*classe em wlc*), 160
weblate admin command
add_suggestions, 384
auto_translate, 384
celery_queues, 385
checkgit, 385
cleanup_ssh_keys, 386
cleanuptrans, 386
commit_pending, 385
commitgit, 385
createadmin, 386
dump_memory, 387
dumpuserdata, 387
import_demo, 387
import_json, 387
import_memory, 388
import_project, 388
importuserdata, 391
importusers, 391
install_addon, 391
list_languages, 391
list_translators, 392
list_versions, 392
loadpo, 392
lock_translation, 392
move_language, 393
pushgit, 393
setupgroups, 394
setuplang, 394
unlock_translation, 393
updatechecks, 394
updategit, 394
WEBLATE_ADDONS
setting, 365
WEBLATE_ADMIN_EMAIL, 167--169
WEBLATE_ADMIN_NAME, 167--169
WEBLATE_ADMIN_PASSWORD, 163, 167--169
WEBLATE_ADMIN_PASSWORD_FILE, 169
WEBLATE_AKISMET_API_KEY, 401
WEBLATE_ALLOWED_HOSTS, 210, 214, 363
WEBLATE_API_RATELIMIT_ANON, 109, 110
WEBLATE_API_RATELIMIT_USER, 109, 110
WEBLATE_AUTH_LDAP_BIND_PASSWORD, 174
WEBLATE_EMAIL_HOST_PASSWORD, 181
WEBLATE_EMAIL_PORT, 180, 181
WEBLATE_EMAIL_USE_SSL, 180, 181
WEBLATE_EMAIL_USE_TLS, 180, 181
WEBLATE_ENABLE_HTTPS, 243
WEBLATE_EXPORTERS
setting, 366
WEBLATE_FORMATS
setting, 366
WEBLATE_GPG_IDENTITY
setting, 367
WEBLATE_LOCALIZE_CDN_PATH, 182
WEBLATE_MACHINERY

setting, 367
WEBLATE_RATELIMIT_ATTEMPTS, 403
WEBLATE_REQUIRE_LOGIN, 362
WEBLATE_SECURE_PROXY_SSL_HEADER, 170
WEBLATE_SERVICE, 167
WEBLATE_SILENCED_SYSTEM_CHECKS, 240
WEBLATE_SITE_DOMAIN, 212, 230, 363
WEBLATE_WORKERS, 167, 183, 184
WeblateConfig (*classe em wlc.config*), 161
WeblateException, 160
WEBSITE_REQUIRED
setting, 367
wlc, 155
módulo, 160
wlc command line option
--author-email, 158
--author-name, 158
--config, 156
--config-section, 156
--convert, 157
--format, 156
--fuzzy, 158
--input, 158
--key, 156
--method, 158
--output, 157
--overwrite, 158
--url, 156
changes, 157
cleanup, 157
commit, 157
download, 157
list-components, 156
list-languages, 156
list-projects, 156
list-translations, 156
lock, 157
lock-status, 157
ls, 157
pull, 157
push, 157
repo, 157
reset, 157
show, 156
stats, 157
unlock, 157
upload, 157
version, 156
wlc.config
módulo, 161
wlc.main
módulo, 161
WPF
file format, 92

X

XLIFF
file format, 82
XML
file format, 95

Y

YAML
file format, 93
YAML Ain't Markup Language
file format, 93