



The Weblate Manual

Release 4.16

Michal Čihař

01 mar. 2023

1	Documentação de usuário	1
1.1	Básico do Weblate	1
1.2	Registro e perfil de usuário	1
1.3	Traduzindo usando o Weblate	10
1.4	Baixando e enviando traduções	20
1.5	Glossário	23
1.6	Verificações e correções	26
1.7	Pesquisando	59
1.8	Fluxos de trabalho de tradução	64
1.9	Perguntas frequentes	68
1.10	Formatos de arquivos suportados	76
1.11	Integração com controle de versão	99
1.12	API REST do Weblate	107
1.13	Weblate Client	155
1.14	API Python do Weblate	160
2	Documentação de administrador	162
2.1	Instruções de configuração	162
2.2	Implantações de Weblate	228
2.3	Atualizando o Weblate	229
2.4	Fazendo backup e movendo o Weblate	238
2.5	Autenticação	246
2.6	Controle de acesso	256
2.7	Projetos de tradução	265
2.8	Definições de idioma	284
2.9	Localização contínua	287
2.10	Licenciando traduções	296
2.11	Processo de tradução	298
2.12	Verificações e correções	304
2.13	Configurando sugestões automáticas	314
2.14	Extensões	325
2.15	Memória de tradução	343
2.16	Configuração	345
2.17	Amostra de configuração	371
2.18	Comandos de gerência	387
2.19	Anúncios	398
2.20	Lista de componentes	401
2.21	Módulos opcionais do Weblate	402
2.22	Personalizando o Weblate	407
2.23	Interface de gerenciamento	410
2.24	Obtendo suporte para o Weblate	418

2.25	Documentos legais	421
3	Documentação de colaborador	423
3.1	Contribuindo para o Weblate	423
3.2	Começando a contribuir código para o Weblate	425
3.3	Código-fonte do Weblate	429
3.4	Depuração do Weblate	430
3.5	Componentes internos do Weblate	432
3.6	Desenvolvendo extensões	433
3.7	Frontend do Weblate	435
3.8	Relatando problemas no Weblate	436
3.9	Conjunto de testes e integração contínua do Weblate	436
3.10	Esquemas de dados	438
3.11	Criando lançamento do Weblate	442
3.12	Segurança e privacidade	443
3.13	Contribuindo para módulos do Weblate	444
3.14	Sobre o Weblate	445
3.15	Licença	445
4	Histórico de alterações	446
4.1	Weblate 4.16	446
4.2	Weblate 4.15.2	446
4.3	Weblate 4.15.1	447
4.4	Weblate 4.15	447
4.5	Weblate 4.14.2	448
4.6	Weblate 4.14.1	448
4.7	Weblate 4.14	449
4.8	Weblate 4.13.1	449
4.9	Weblate 4.13	449
4.10	Weblate 4.12.2	450
4.11	Weblate 4.12.1	450
4.12	Weblate 4.12	451
4.13	Weblate 4.11.2	451
4.14	Weblate 4.11.1	451
4.15	Weblate 4.11	452
4.16	Weblate 4.10.1	452
4.17	Weblate 4.10	453
4.18	Weblate 4.9.1	453
4.19	Weblate 4.9	453
4.20	Weblate 4.8.1	454
4.21	Weblate 4.8	455
4.22	Weblate 4.7.2	455
4.23	Weblate 4.7.1	455
4.24	Weblate 4.7	456
4.25	Weblate 4.6.2	456
4.26	Weblate 4.6.1	457
4.27	Weblate 4.6	457
4.28	Weblate 4.5.3	458
4.29	Weblate 4.5.2	458
4.30	Weblate 4.5.1	458
4.31	Weblate 4.5	459
4.32	Weblate 4.4.2	459
4.33	Weblate 4.4.1	460
4.34	Weblate 4.4	460
4.35	Weblate 4.3.2	461
4.36	Weblate 4.3.1	461
4.37	Weblate 4.3	462
4.38	Weblate 4.2.2	463

4.39	Weblate 4.2.1	463
4.40	Weblate 4.2	463
4.41	Weblate 4.1.1	464
4.42	Weblate 4.1	464
4.43	Weblate 4.0.4	465
4.44	Weblate 4.0.3	466
4.45	Weblate 4.0.2	466
4.46	Weblate 4.0.1	466
4.47	Weblate 4.0	467
4.48	Série 3.x do Weblate	467
4.49	Série 2.x do Weblate	479
4.50	Série 1.x do Weblate	490
4.51	Série 0.x do Weblate	494
Índice de Módulos Python		498
HTTP Routing Table		499
Índice		502

1.1 Básico do Weblate

1.1.1 Estrutura de projetos e componentes

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

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

Ver também:

`../devel/integration`

1.2 Registro e perfil de usuário

1.2.1 Registro

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

Você pode se registrar seguindo alguns passos simples:

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

1.2.2 Painei

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

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

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

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a user profile section with a 'Your profile' link and a menu of tabs: 'Languages', 'Preferences' (which is active), 'Notifications', 'Account', 'Profile', 'Teams', 'Licenses', 'Audit log', and 'API access'. The 'Preferences' panel is open, showing several settings:

- ☐ Hide completed translations on the dashboard
- Translation editor mode**: A dropdown menu set to 'Full editor'.
- Zen editor mode**: A dropdown menu set to 'Top to bottom'.
- Number of nearby strings**: A text input field set to '15'. Below it, a note says 'Number of nearby strings to show in each direction in the full editor.'
- ☒ Show secondary translations in the Zen mode
- ☐ Hide source if a secondary translation exists
- Editor link**: A text input field. Below it, a note says 'Enter a custom URL to be used as link to the source code. You can use {{branch}} for branch, {{filename}} and {{line}} as filename and line placeholders.'
- Special characters**: A text input field. Below it, a note says 'You can specify additional special visual keyboard characters to be shown while translating. It can be useful for characters you use frequently, but are hard to type on your keyboard.'
- Default dashboard view**: Two radio buttons, 'Watched translations' (which is selected) and 'Suggested translations'.

At the bottom of the preferences panel is a 'Save' button.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

O menu tem estas opções:

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

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

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

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

1.2.3 Perfil do usuário

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

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

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

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

Idiomas

1.2.4 Idioma da interface

Escolha o idioma que você deseja exibir na UI.

Idiomas traduzidos

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

Weblate Dashboard Projects Languages Checks							
Dashboard							
Watched translations 13		Suggested translations 5		Insights		Search	
Translation	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
WeblateOrg/Android — Czech MIT	76%	3	3	10			
WeblateOrg/Django — Czech GPL-3.0	96%	1	12	69	4		
WeblateOrg/Django — Hebrew GPL-3.0	92%	2	15	80			
WeblateOrg/Django — Hungarian GPL-3.0	69%	8	109	671	1		
WeblateOrg/Djangojs — Hungarian GPL-3.0	96%	2	6	28			
WeblateOrg/Djangojs — Hebrew GPL-3.0	✓						
WeblateOrg/Djangojs — Czech GPL-3.0	✓						
WeblateOrg/Language names — Czech GPL-3.0	✓						
WeblateOrg/Language names — Hungarian GPL-3.0	81%	4	5	32			
WeblateOrg/Language names — Hebrew GPL-3.0	✓						
WeblateOrg/WeblateOrg — Hungarian GPL-3.0	✓						
WeblateOrg/WeblateOrg — Czech GPL-3.0	✓						
WeblateOrg/WeblateOrg — Hebrew GPL-3.0	✓						

Idiomas secundários

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

The screenshot displays the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the breadcrumb 'WeblateOrg / Django / Czech / Translate' is shown. The main area is divided into a 'Translation' editor and a right sidebar. The editor has input fields for 'Hebrew' (containing 'קבצים'), 'English' (containing 'Files'), and 'Czech' (containing 'Soubory'). It includes a 'Needs editing' checkbox and buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. Below the editor are tabs for 'Nearby strings' (16), 'Comments', 'Automatic suggestions', 'Other languages' (3), and 'History'. A table below shows the 'Language' and 'Target string' for Hebrew, Hungarian ('Fájlok'), and English ('Files'). The right sidebar contains a 'Glossary' section, a 'String information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

1.2.5 Preferências

Visão padrão do painel

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

Ver também:

Lista de componentes

Link do editor

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

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

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

Ver também:

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

Caracteres especiais

Caracteres especiais adicionais para incluir no *Teclado visual*.

1.2.6 Notificações

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


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

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

No caso de *Observar automaticamente projetos em contribuição* estar habilitado, você começará automaticamente a ver os projetos ao traduzir um texto. O valor padrão depende de `DEFAULT_AUTO_WATCH`.

Nota: Você não receberá notificações para suas próprias ações.

Dica: Sending out notifications is limited, you will not receive more than 1000 e-mails per day. Any further notifications for you will be discarded.

 Weblate
 Dashboard
 Projects
 Languages
 Checks

Your profile
 Languages
 Preferences
 Notifications
 Account
 Profile
 Teams
 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 4.16
 About Weblate
 Legal
 Contact
 Documentation
 Donate to Weblate

1.2.7 Conta

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

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

Webplate

Dashboard

Projects ▾

Languages ▾

Checks ▾

+

▾

▾

...

Your profile

Languages

Preferences

Notifications

Account

Profile

Teams

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.

Commit e-mail


Use account e-mail address

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

1.2.8 Perfil

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

O e-mail de commit será usado em vez do e-mail da sua conta em commits de controle de versão. Use isso para evitar o vazamento de seu e-mail real lá. Esteja ciente de que o uso de e-mail diferente pode desconectar suas contribuições em outros servidores (por exemplo, suas contribuições não serão mais vinculadas ao seu perfil no GitHub). O e-mail privado pode ser ativado em todo o site usando `PRIVATE_COMMIT_EMAIL_OPT_IN`.

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

1.2.9 Licenças

1.2.10 Acesso a API

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

1.2.11 Registro de auditoria

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

Ver também:

Executando por trás de um proxy reverso

1.3 Traduzindo usando o Weblate

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

No geral, há dois modos de tradução:

- O projeto aceita traduções diretas
- O projeto aceita apenas sugestões, que são validadas automaticamente uma vez que um número definido de votos é alcançado

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

Opções para a visibilidade do projeto de tradução:

- Publicamente visível
- Visível apenas para um certo grupo de tradutores

Ver também:

Controle de acesso, Fluxos de trabalho de tradução

1.3.1 Projetos de tradução

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

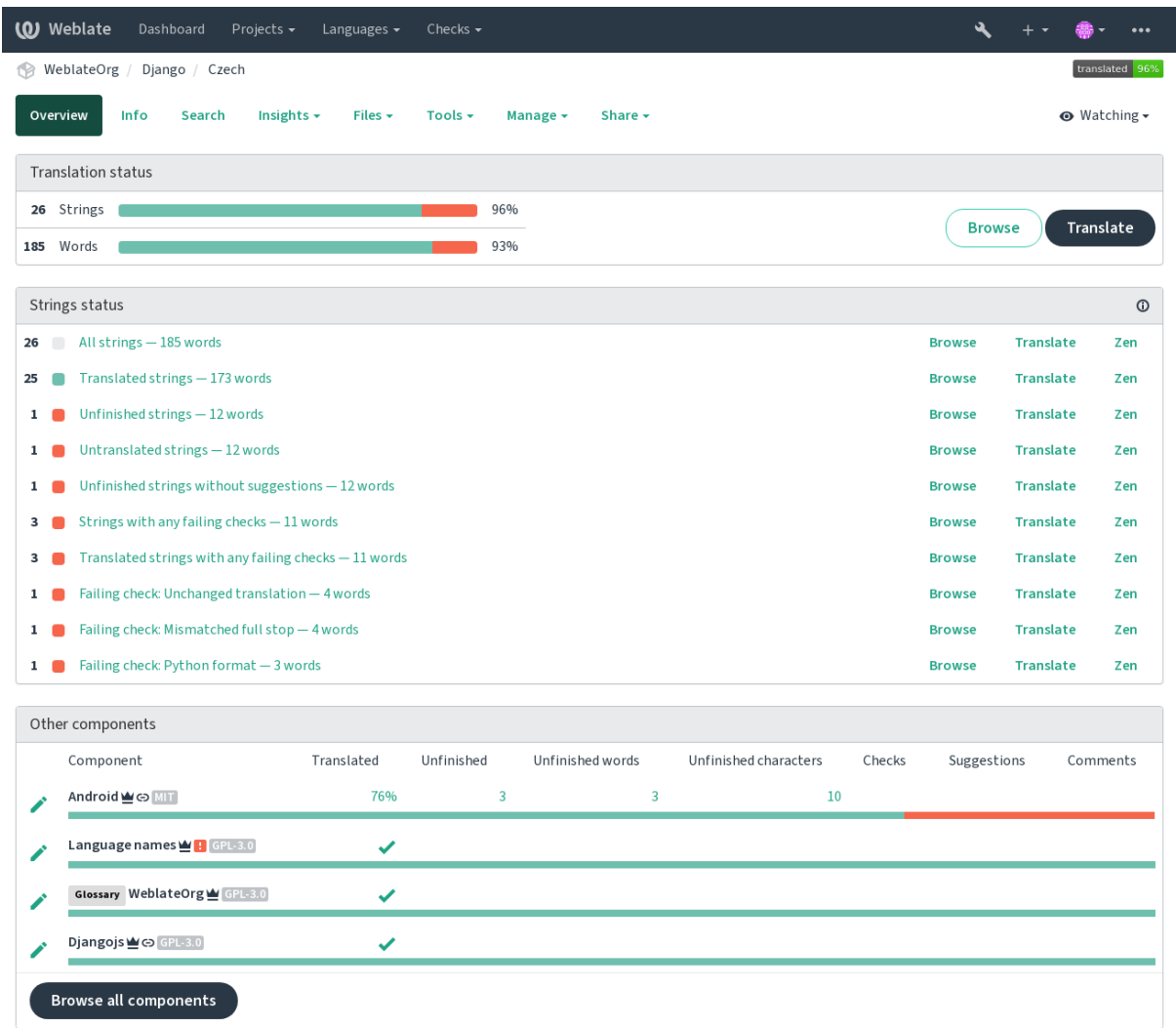
Component	Translated	Unfinished	Unfinished words	Checks	Suggestions	Comments
Android	79%	30	30	3		
Language names	95%	4	5			
Glossary WeblateOrg	100%	0	0	0		

[Add new translation component](#)

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.3.2 Links de tradução

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



Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.3.3 Sugestões

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

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

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

1.3.4 Comentários

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

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

Ver também:

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

1.3.5 Variantes

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

Ver também:

variants, *Variantes*

1.3.6 Etiquetas

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

As etiquetas são usadas pelo Weblate:

Traduzido automaticamente

O texto foi traduzido usando *Tradução automática*.

Fonte precisa de revisão

O texto foi marcado para revisão usando *Revisões de textos fonte*.

Ver também:

labels

1.3.7 Traduzindo

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

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

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

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

Plurais

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

O Weblate tem total suporte a cada uma dessas formas, em cada respectivo idioma (traduzindo cada plural separadamente). O número de campos e como ele é usado no aplicativo ou projeto traduzido depende da forma de plural configurada. Weblate mostra as informações básicas, e [Language Plural Rules](#) do Unicode Consortium tem informações mais detalhadas.

Ver também:

Fórmula de plural

The screenshot displays the Weblate web interface for translating a string. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The main area shows the translation process for a string in Czech. The English string is "%(count)s word". The Czech translations are provided for "One", "Few", and "Many" forms, all using the placeholder "%(count)s slovo". The plural formula is shown as $(n=1) ? 0 : (n>2 \ \&\& \ n<=4) ? 1 : 2$. The interface includes buttons for "Save and continue", "Save and stay", "Suggest", and "Skip". A sidebar on the right contains a Glossary, String information, and a Comments section. The bottom of the page features a footer with links for About Weblate, Legal, Contact, Documentation, and Donate to Weblate.

Traduções alternativas

Novo na versão 4.13.

Nota: Isso só tem suporte atualmente em *Arquivo CSV com vários valores*.

Com alguns formatos, é possível ter mais traduções para um único texto. Você pode adicionar mais traduções alternativas usando o menu *Ferramentas*. Quaisquer traduções alternativas em branco serão removidas automaticamente ao salvar.

Atalhos de teclado

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

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

Atalho de teclado	Descrição
Alt+Home	Navega para a primeira tradução na pesquisa atual.
Alt+End	Navega para a última tradução na pesquisa atual.
Alt+PageUp ou Ctrl+↑ ou Alt+↑ ou Cmd+↑	Navega para a tradução anterior na pesquisa atual.
Alt+PageDown ou Ctrl+↓ ou Alt+↓ ou Cmd+↓	Navega para a próxima tradução na pesquisa atual.
Ctrl+Enter ou Cmd+Enter	Envia o formulário atual; isso é o mesmo que pressionar <i>Salvar e continuar</i> enquanto edita a tradução.
Ctrl+Shift+Enter ou Cmd+Shift+Enter	Desmarca a tradução como precisando de edição e a envia.
Alt+Enter ou Option+Enter	Envia o texto como sugestão; isso é o mesmo que pressionar <i>Sugerir</i> enquanto edita a tradução.
Ctrl+E ou Cmd+E	Muda o foco para o editor de tradução.
Ctrl+U ou Cmd+U	Muda o foco para o editor de comentários.
Ctrl+M ou Cmd+M	Mostra a aba <i>Sugestões automáticas</i> , consulte <i>Sugestões automáticas</i> .
Ctrl+1 até Ctrl+9 ou Cmd+1 até Cmd+9	Copia objetos colocáveis de determinado número do texto fonte.
Ctrl+M+1 até 9 ou Cmd+M+1 até 9	Copia a tradução de máquina do número dado para a tradução atual.
Ctrl+I+1 até 9 ou Cmd+I+1 até 9	Ignora o item na lista de verificações com falha.
Ctrl+J ou Cmd+J	Mostra a aba de <i>Textos próximos</i> .
Ctrl+S ou Cmd+S	Muda o foco para o campo de pesquisa.
Ctrl+O ou Cmd+O	Copia o texto fonte.
Ctrl+Y ou Cmd+Y	Marca ou desmarca a caixa de seleção de <i>Necessita edição</i> .

Teclado visual

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

Os símbolos mostrados são apresentados em três categorias:

- *Caracteres especiais* configurados pelo usuário definidos em *Perfil do usuário*
- Caracteres por idioma fornecidos pelo Weblate (por exemplo, citações ou caracteres específicos RTL)
- Caracteres configurados usando `SPECIAL_CHARS`

The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below it, a breadcrumb trail shows 'WeblateOrg / Django / Hebrew / Translate'. The main area is the translation editor. It has a 'Translation' tab with an 'English' source string 'Files' and a 'Hebrew' target string field. Below the field are buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. To the right of the field is a 'Visual keyboard' with various symbols. Below the editing area is a table of 'Nearby strings' for Czech, Hungarian, and English. On the right side, there is a 'Glossary' section and a 'String information' sidebar with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'.

Contexto da tradução

Esta descrição contextual fornece informações relacionadas sobre o texto atual.

Atributos do texto

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

Capturas de tela

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

Textos próximos

Exibe mensagens próximas do arquivo de tradução. Estas também são geralmente usadas em um contexto semelhante e se mostram úteis para manter a tradução consistente.

Outras ocorrências

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

Memória de tradução

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

Glossário

Exibe termos do glossário do projeto usados na mensagem atual.

Alterações recentes

Lista de pessoas que mudaram esta mensagem recentemente usando Weblate.

Projeto

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

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

Histórico de tradução

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

Comprimento do texto traduzido

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

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

1.3.8 Sugestões automáticas

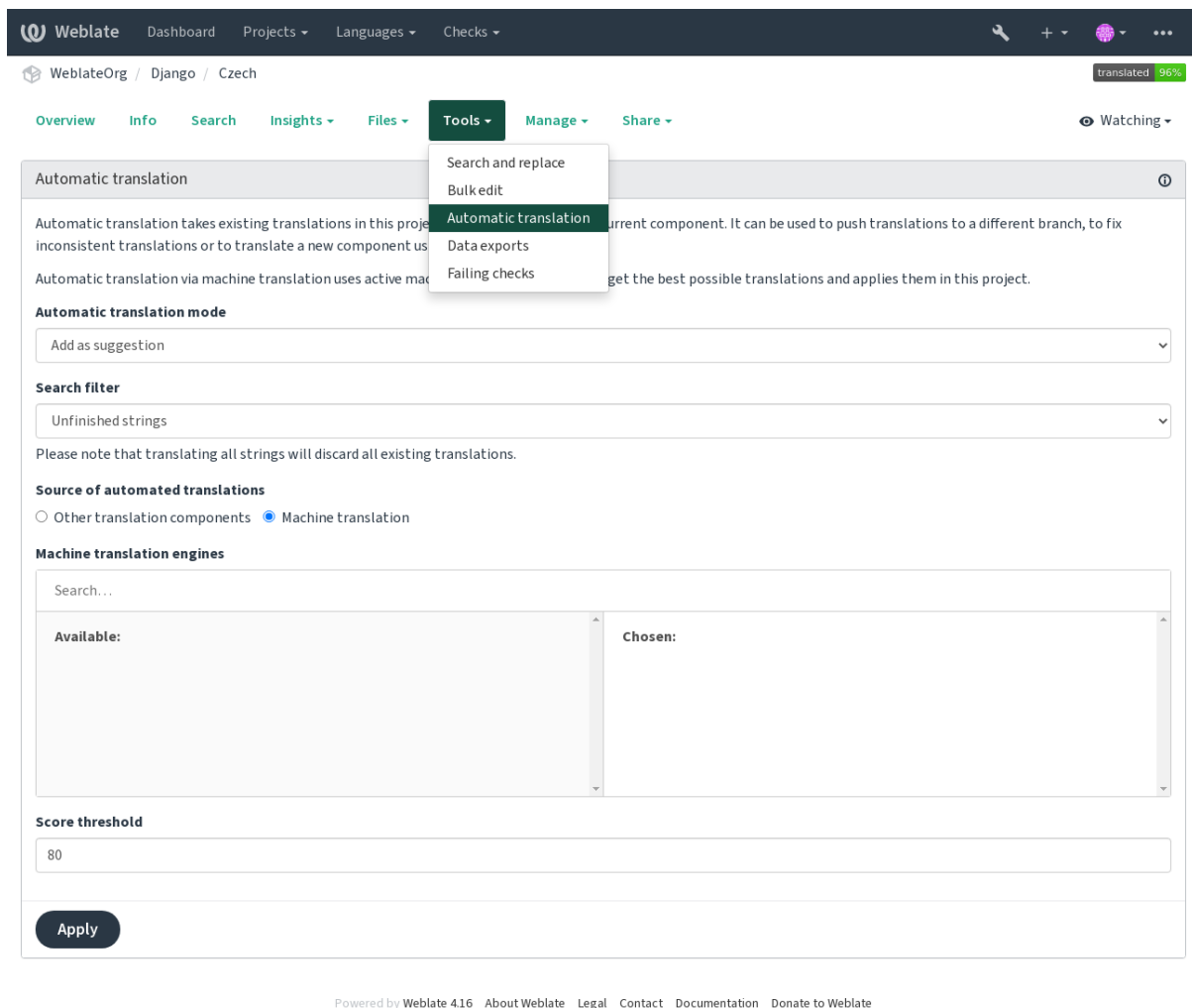
Baseado na configuração e seu idioma traduzido, o Weblate fornece sugestões de várias ferramentas de tradução e memória-tradução. Todas as traduções de máquina estão disponíveis em uma única aba de cada página de tradução.

Ver também:

Você pode encontrar a lista de ferramentas suportadas em [Configurando sugestões automáticas](#).

1.3.9 Tradução automática

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



Dois modos de operação são possíveis:

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

Você também pode escolher quais textos devem ser traduzidos automaticamente.

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

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

Os textos traduzidos automaticamente são etiquetados como *Traduzido automaticamente*.

Ver também:

Mantendo traduções iguais entre componentes

1.3.10 Limitação de taxa

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

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

1.3.11 Pesquisar e substituir

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

Dica: Não se preocupe em bagunçar os textos. Este é um processo de duas etapas que mostra uma visualização dos textos editados antes que a mudança real seja confirmada.

1.3.12 Editor em massa

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

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

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

Ver também:

[Extensão de edição em massa](#)

1.3.13 Visualização em matriz

Para comparar diferentes idiomas de forma eficiente, você 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 sua seleção, depois disso, você 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 habilitado clicando 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*.

Você pode selecionar o editor Zen como seu editor padrão usando a aba [Preferências](#) em seu [Perfil do usuário](#). Aqui você também pode escolher entre ter traduções listadas *De cima para baixo* ou *Lado a lado* dependendo de sua preferência pessoal.

1.4 Baixando e enviando traduções

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

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

1.4.1 Baixando traduções

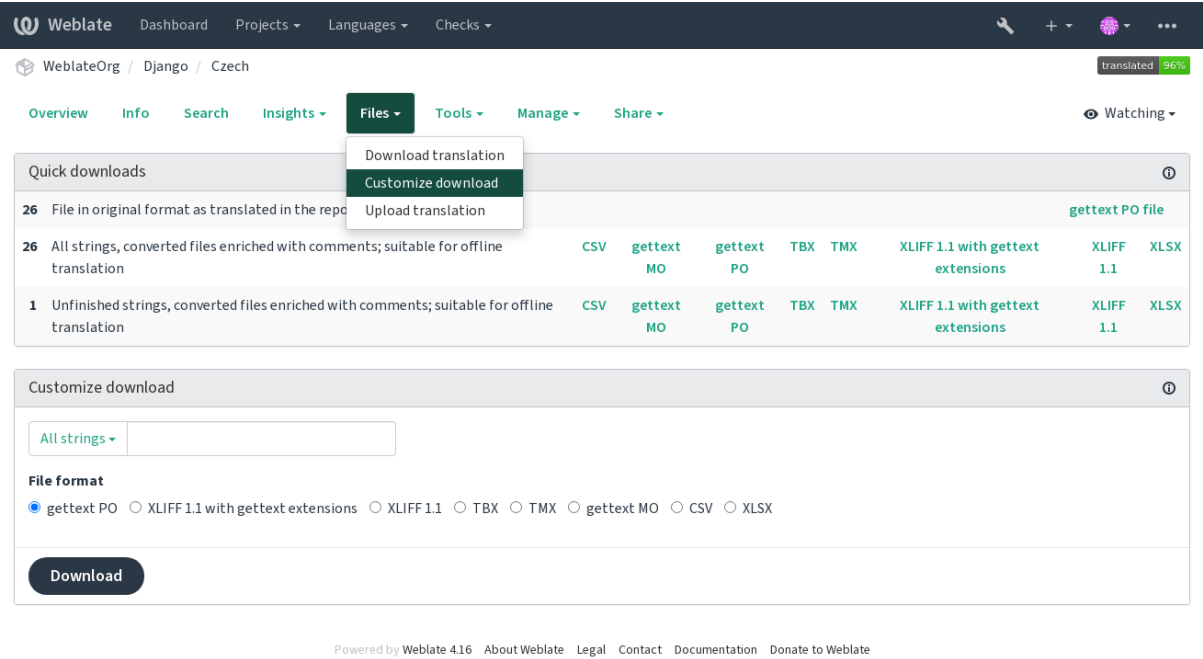
A partir do painel do projeto ou componente, arquivos traduzíveis podem ser baixados no menu *Arquivos*.

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

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

- gettext PO
- XLIFF com extensões gettext
- XLIFF 1.1
- TermBase eXchange
- Translation Memory eXchange
- gettext MO (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 arquivos convertidos difere com base nos recursos de formato de arquivo, você pode encontrar uma visão geral em *Capacidades dos tipos de tradução*.



Ver também:

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

1.4.2 Enviando traduções

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

Upload

The uploaded file will be merged with the current file. Use the dropdown menu if you want to overwrite already translated strings.

File

No file chosen

File upload mode

☐ Add as translation

☐ Add as suggestion

☐ Add as translation needing edit

☐ Replace existing translation file

Processing of "Needs editing" strings

Conflict handling

Whether to overwrite existing translations if the string is already translated.

Author name

Author e-mail

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Formatos de arquivos suportados

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

Ver também:

Formatos de arquivos suportados, Baixando e enviando traduções

Métodos de importação

Estas são as opções apresentadas ao enviar arquivos 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 arquivo carregado e nenhum conteúdo adicional.

Adicionar como sugestão (**suggest**)

Os textos importados são adicionados como sugestões, faça isso quando quiser ter seus textos enviados revisados.

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

Adicionar como tradução que necessita edição (**fuzzy**)

Os textos importados são adicionados como traduções que necessitam de edição. Isso pode ser útil quando você quer que as traduções sejam usadas, mas também revisadas.

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

Substituir arquivo tradução existente (replace)

O arquivo existente é substituído por novo conteúdo. Isso pode levar à perda de traduções existentes, use com cuidado.

Atualizar textos fonte (source)

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

Esta opção é suportada apenas para alguns formatos de arquivo.

Adicionar novos textos (add)

Adiciona novos textos à tradução. Pula a que já existe.

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

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

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

Ver também:

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

Gestão de conflitos

Define como lidar com textos enviados que já são traduzidos.

Textos necessitando de edição

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

Substituindo autoria

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

1.5 Glossário

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

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

1.5.1 Gerenciando glossários

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

Use qualquer componente como um glossário ativando *Usar como um glossário*. Você pode criar vários glossários para um projeto.

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

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

Translation status

2 Strings	100%
3 Words	100%

Buttons: Add new glossary term, Browse, Translate

Strings status

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

Other components

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

Buttons: Browse all components

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Você pode navegar por todos os termos do glossário:

Navigation: All strings, Source string, Add new glossary term

English	Czech
machine translation	strojový překlad
project	projekt

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

ou editá-los como qualquer tradução.

1.5.2 Termos de glossário

Os termos de glossário são traduzidos da mesma forma que os textos comuns. Você pode alternar recursos adicionais utilizando o menu *Ferramenta* para cada termo.

The screenshot displays the Weblate web interface for managing a glossary. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The main content area is titled 'Glossary term' and shows a term being edited. The 'English' field contains 'project', and the 'Czech' field contains 'projekt'. Below these fields is an 'Explanation' section. The interface also features a 'Tools' dropdown menu with options such as 'Delete string', 'Mark as untranslatable', 'Mark as forbidden translation', 'Mark as terminology', and 'Add variant of this string'. The right sidebar provides additional context, including 'String information' and 'Translation file' details.

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Termos não traduzíveis

Novo na versão 4.5.

Marcar certas traduções de termos do glossário como `read-only` por meio de edição massa, digitação do marcador ou uso de *Ferramentas* ↓ *Marcar como intraduzível* significa que não podem ser traduzidas. Use-o para nomes de marcas ou outros termos que não devem ser alterados em outros idiomas. Esses termos são destacados visualmente na barra lateral do glossário.

Ver também:

Personalizando comportamento usando marcadores

Traduções proibidas

Novo na versão 4.5.

Marcar certas traduções de termos do glossário como `forbidden` por meio de edição massa, digitação do marcador ou uso de *Ferramentas* ↓ *Marcar como tradução proibida* significa que **não** devem ser usadas. Use isto para esclarecer a tradução quando algumas palavras são ambíguas ou podem ter significados inesperados.

Ver também:

Personalizando comportamento usando marcadores

Terminologia

Novo na versão 4.5.

Marcar certos termos do glossário como `terminology` por meio de edição massa, digitação do marcador ou uso de *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.

Ver também:

Personalizando comportamento usando marcadores

Variantes

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

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

Ver também:

variants

1.6 Verificações e correções

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

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

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 1

Translation

English

Singular

%(count)s word

Plural

%(count)s words

Czech, One

Czech, Few

několik slov

Czech, Many

%(count)s slov

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

☐ Needs editing

Save and continue

Save and stay

Suggest

Skip

Nearby strings 20

Comments

Automatic suggestions

Other languages 3

History

New comment

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

Scope

Translation comment, discussions with other translators

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

New comment

You can use Markdown and mention users by @username.

Save

Things to check

Python format 1

Following format strings are missing: %(count)s

Dismiss

☐ For all languages

Missing plurals 2

Some plural forms are untranslated

Dismiss

☐ For all languages

Glossary

EnglishCzech

No related strings found in the glossary.

Add term to glossary

String information

Screenshot context

No screenshot currently associated.

Add screenshot

Explanation

No explanation currently provided.

Labels

No labels currently set.

Flags

python-format

Source string location

weblate/templates/translation.html:149

String age

10 seconds ago

Source string age

10 seconds ago

Translation file

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

Powered by Weblate 4.16

About Weblate

Legal

Contact

Documentation

Donate to Weblate

1.6. Verificações e correções

27

1.6.1 Correções automáticas

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

Ver também:

AUTOFIX_LIST

1.6.2 Verificações de qualidade

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

Ver também:

CHECK_LIST, *Personalizando comportamento usando marcadores*

1.6.3 Verificações de tradução

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

Marcação de BBCode

Resumo

O BBCode na tradução não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.BBCodeCheck`

Identificador da verificação

`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 sejam encontrados na tradução.

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

Há palavras duplicadas de forma consecutiva

Novo na versão 4.1.

Resumo

O texto contém a mesma palavra repetida uma ao lado da outra:

Escopo

textos traduzidos

Classe da verificação

`weblate.checks duplicate.DuplicateCheck`

Identificador da verificação

`duplicate`

Marcador para ignorar

`ignore-duplicate`

Verifica se não ocorrem palavras duplicadas consecutivas em uma tradução. Isso geralmente indica um erro na tradução.

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

Não segue o glossário

Novo na versão 4.5.

Resumo

A tradução não segue termos definidos em um glossário.

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.glossary.GlossaryCheck`

Identificador da verificação

`check_glossary`

Marcador para habilitar

`check-glossary`

Marcador para ignorar

`ignore-check-glossary`

Esta verificação tem de ser ativada usando o marcador `check-glossary` (veja [Personalizando comportamento usando marcadores](#)). Considere o seguinte antes de habilitá-la:

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

Ver também:

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

Espaço duplo

Resumo

A tradução contém espaço duplo

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.DoubleSpaceCheck`

Identificador da verificação

`double_space`

Marcador para ignorar

`ignore-double-space`

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

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

Textos formatados

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

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

Isso pode ser feito por unidade (veja [Informações adicionais sobre textos fonte](#)) ou na [Configuração de componente](#). Tê-lo definido por componente é mais simples, mas pode levar a falsos positivos no caso de o texto não ser interpretado como um texto de formatação, mas a sintaxe de textos de formato passa a ser usada.

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

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

The screenshot shows the Weblate web interface for a project named 'Django' in the 'Czech' language. The main panel displays the translation of the English string '%(count)s word'. The Czech translations are shown for different grammatical forms: Singular, Plural, Czech, One, Czech, Few, and Czech, Many. The right sidebar provides additional information about the string, including its context, explanation, labels, flags, source location, and age. The bottom section shows a list of nearby strings and a detailed view of a specific string update.

Translation Panel:

- English:** Singular: '%(count)s word', Plural: '%(count)s words'
- Czech, One:** '%(count)s slovo'
- Czech, Few:** '%(count)s slova'
- Czech, Many:** '%(count)s slov'
- Plural formula:** (n==1) ? 0 : (n>=2 && n<=4) ? 1 : 2
- Needs editing:** ☐
- Buttons:** Save and continue, Save and stay, Suggest, Skip

String information Panel:

- Screenshot context:** No screenshot currently associated.
- 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

Nearby strings: 20

Comments: Automatic suggestions, Other languages: 3, History

String update details:

- English:** Singular: '%(count)s word', Plural: '%(count)s words'
- Czech:** 48 characters edited, Current translation, Translated
- One:** '%(count)s slovo'
- Few:** '%(count)s slova'
- Many:** '%(count)s slov'
- 6 seconds ago**

[Browse all string changes](#)

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Texto de interpolação AngularJS

Resumo

O texto de interpolação AngularJS não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.angularjs.AngularJSInterpolationCheck`

Identificador da verificação

`angularjs_format`

Marcador para habilitar

`angularjs-format`

Marcador para ignorar

`ignore-angularjs-format`

Exemplo de texto de formato nomeado

Seu saldo é `{{amount}}` `{{ currency }}`

Ver também:

Textos formatados, [Interpolação de texto do AngularJS](#)

Formato C

Resumo

O texto de formato C não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.CFormatCheck`

Identificador da verificação

`c_format`

Marcador para habilitar

`c-format`

Marcador para ignorar

`ignore-c-format`

Exemplo de texto de formato simples

Há `%d` maçãs

Exemplo de texto de formato de posição

Seu saldo é `%1$d` `%2$s`

Ver também:

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

Formato C#

Resumo

O texto de formato C# não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.CSharpFormatCheck`

Identificador da verificação

`c_sharp_format`

Marcador para habilitar

`c-sharp-format`

Marcador para ignorar

`ignore-c-sharp-format`

Exemplo de texto de formato de posição

Há `{0}` maçãs

Ver também:

Textos formatados, *Formato de string do C#*

Literais de modelo de ECMAScript**Resumo**

Literais de modelo ECMAScript não correspondem com a fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.ESTemplateLiteralsCheck`

Identificador da verificação

`es_format`

Marcador para habilitar

`es-format`

Marcador para ignorar

`ignore-es-format`

Exemplo de interpolação

Há `${number}` maçãs

Ver também:

Textos formatados, *Template strings*

Interpolação de i18next

Novo na versão 4.0.

Resumo

A interpolação de i18next não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.I18NextInterpolationCheck`

Identificador da verificação

`i18next_interpolation`

Marcador para habilitar

`i18next-interpolation`

Marcador para ignorar

`ignore-i18next-interpolation`

Exemplo de interpolação

Há `{{number}}` maçãs

Exemplo de aninhamento

Há `$t(number)` maçãs

Ver também:

Textos formatados, *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.

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.icu.ICUMessageFormatCheck`

Identificador da verificação

`icu_message_format`

Marcador para habilitar

`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. Você pode configurar o comportamento desta verificação usando `icu-flags:*`, ou optando pelo suporte XML ou desabilitando certas sub-verificações. Por exemplo, a seguinte sinalização ativa o suporte a XML enquanto desativa a validação de submensagens plurais:

```
icu-message-format, icu-flags:xml:-plural_selectors
```

<code>xml</code>	Habilita 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.
<code>strict-xml</code>	Habilita o suporte para tags XML estritas. Todos os caracteres < devem ser escapados se não fizerem parte de uma tag.
<code>-highlight</code>	Desabilita o realce de espaços reservados no editor.
<code>-require_oth</code>	Desabilita a exigência de submensagens ter um seletor <code>other</code> .
<code>-submessage_</code>	Ignora a verificação de que os seletores de submensagem correspondem à fonte.
<code>-types</code>	Ignora a verificação de que os tipos de espaço reservado correspondem à fonte.
<code>-extra</code>	Ignora a verificação de nenhum espaço reservado está presente que não estava presente no texto fonte.
<code>-missing</code>	Ignora a verificação de que não falta nenhum espaço reservado que estava presente no texto fonte.

Além disso, quando `strict-xml` não está habilitado, mas `xml` está habilitado, você 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::`:

```
icu-message-format, icu-flags:xml, icu-tag-prefix:"x:"
```

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

Ver também:

Sintaxe de MessageFormat do ICU, *Textos formatados*, ICU: Formatação de mensagens, Format.JS: sintaxe de mensagens

Formato Java

Resumo

O texto de formato Java não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.JavaFormatCheck`

Identificador da verificação

`java_printf_format`

Marcador para habilitar

`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

Seu saldo é %1\$d %2\$s

Alterado na versão 4.14: Isso costumava ser alternado pelo marcador `java-format`, foi alterado para consistência com GNU `gettext`.

Ver também:

Textos formatados, [Java Format Strings](#)

MessageFormat do Java

Resumo

O texto MessageFormat do Java não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.JavaMessageFormatCheck`

Identificador da verificação

`java_format`

Marcador para habilitar incondicionalmente

`java-format`

Marcador para habilitar autodetecção

`auto-java-messageformat` habilita 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: Isso costumava ser alternado pelo marcador `java-messageformat`, foi alterado para consistência com GNU `gettext`.

Essa verificação valida se o texto de formato é válido para a classe Java `MessageFormat`. Além de corresponder aos textos de formato nas chaves, ele também verifica as aspas simples, pois elas têm um significado especial. Sempre

que escrever aspas simples, deve-se escrever como ' '. Quando não emparelhado, é tratado como início de aspas e não será mostrado ao renderizar o texto.

Ver também:

Textos formatados, [Java MessageFormat](#)

Formato JavaScript

Resumo

O texto de formato JavaScript não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.JavaScriptFormatCheck`

Identificador da verificação

`javascript_format`

Marcador para habilitar

`javascript-format`

Marcador para ignorar

`ignore-javascript-format`

Exemplo de texto de formato simples

Há %d maçãs

Ver também:

Textos formatados, [JavaScript formatting strings](#)

Formato Lua

Resumo

O texto de formato Lua não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.LuaFormatCheck`

Identificador da verificação

`lua_format`

Marcador para habilitar

`lua-format`

Marcador para ignorar

`ignore-lua-format`

Exemplo de texto de formato simples

Há %d maçãs

Ver também:

Textos formatados, [Lua formatting strings](#)

Formato de Object Pascal

Resumo

O texto de formato Object Pascal não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.ObjectPascalFormatCheck`

Identificador da verificação

`object_pascal_format`

Marcador para habilitar

`object-pascal-format`

Marcador para ignorar

`ignore-object-pascal-format`

Exemplo de texto de formato simples

Há %d maçãs

Ver também:

Textos formatados, *Object Pascal formatting strings*, *Free Pascal formatting strings* *Delphi formatting strings*

Espaços reservados de porcentagem

Novo na versão 4.0.

Resumo

Os espaços reservados de porcentagem não correspondem à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.PercentPlaceholdersCheck`

Identificador da verificação

`percent_placeholders`

Marcador para habilitar

`percent-placeholders`

Marcador para ignorar

`ignore-percent-placeholders`

Exemplo de texto de formato simples

Há %number% maçãs

Ver também:

Textos formatados,

Formato Perl

Resumo

O texto de formato Perl não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.PperlFormatCheck`

Identificador da verificação

`perl_format`

Marcador para habilitar

`perl-format`

Marcador para ignorar

`ignore-perl-format`

Exemplo de texto de formato simples

Há %d maçãs

Exemplo de texto de formato de posição

Seu saldo é %1\$d %2\$s

Ver também:

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

Formato PHP

Resumo

O texto de formato PHP não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.PHPFormatCheck`

Identificador da verificação

`php_format`

Marcador para habilitar

`php-format`

Marcador para ignorar

`ignore-php-format`

Exemplo de texto de formato simples

Há %d maçãs

Exemplo de texto de formato de posição

Seu saldo é %1\$d %2\$s

Ver também:

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

Formato de chaves Python

Resumo

O texto de formato de chaves Python não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.PythonBraceFormatCheck`

Identificador da verificação

`python_brace_format`

Marcador para habilitar

`python-brace-format`

Marcador para ignorar

`ignore-python-brace-format`

Texto de formato simples

Há {} maçãs

Exemplo de texto de formato nomeado

Seu saldo é {amount} {currency}

Ver também:

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

Formato Python

Resumo

O texto de formato Python não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.PythonFormatCheck`

Identificador da verificação

`python_format`

Marcador para habilitar

`python-format`

Marcador para ignorar

`ignore-python-format`

Texto de formato simples

Há %d maçãs

Exemplo de texto de formato nomeado

Seu saldo é %(amount)d %(currency)s

Ver também:

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

Formato Qt

Resumo

O texto de formato Qt não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.qt.QtFormatCheck`

Identificador da verificação

`qt_format`

Marcador para habilitar

`qt-format`

Marcador para ignorar

`ignore-qt-format`

Exemplo de texto de formato de posição

Há %1 maçãs

Ver também:

Textos formatados, [Qt QString::arg\(\)](#)

Formato de plural Qt

Resumo

O texto de formato Qt não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.qt.QtPluralCheck`

Identificador da verificação

`qt_plural_format`

Marcador para habilitar

`qt-plural-format`

Marcador para ignorar

`ignore-qt-plural-format`

Exemplo de texto de formato de plural

Há %Ln maçã(s)

Ver também:

Textos formatados, [Guia de i18n do Qt](#)

Formato Ruby

Resumo

O texto de formato Ruby não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.ruby.RubyFormatCheck`

Identificador da verificação

`ruby_format`

Marcador para habilitar

`ruby-format`

Marcador para ignorar

`ignore-ruby-format`

Exemplo de texto de formato simples

Há %d maçãs

Exemplo de texto de formato de posição

Seu saldo é %1\$f %2\$s

Exemplo de texto de formato nomeado

Seu saldo é %+.2<amount>f %<currency>s

Texto do modelo nomeado

Seu saldo é %{amount} %{currency}

Ver também:

Textos formatados, [Ruby Kernel#sprintf](#)

Formato Scheme

Resumo

O texto de formato Scheme não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.SchemeFormatCheck`

Identificador da verificação

`scheme_format`

Marcador para habilitar

`scheme-format`

Marcador para ignorar

`ignore-scheme-format`

Exemplo de texto de formato simples

Há ~d maçãs

Ver também:

Textos formatados, [Srifi 28](#), [formato de Scheme Chicken](#), [Saída formatada de Guile Scheme](#)

Formatação Vue I18n

Resumo

A formatação Vue I18n não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.format.VueFormattingCheck`

Identificador da verificação

`vue_format`

Marcador para habilitar

`vue-format`

Marcador para ignorar

`ignore-vue-format`

Formatação nomeada

Há `{count}` maçãs

Formatação Rails i18n

Há `%{count}` maçãs

Mensagens da localidade vinculada

`@:message.dio @:message.the_world!`

Ver também:

Textos formatados, *Formatação Vue I18n*, *Mensagens de localidade Vue I18n*

Foi traduzido

Resumo

Este texto já foi traduzido no passado

Escopo

todos os textos

Classe da verificação

`weblate.checks.consistency.TranslatedCheck`

Identificador da verificação

`translated`

Marcador para ignorar

`ignore-translated`

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

Inconsistente

Resumo

Este texto tem mais de uma tradução neste projeto ou não está traduzido em alguns componentes.

Escopo

todos os textos

Classe da verificação

`weblate.checks.consistency.ConsistencyCheck`

Identificador da verificação`inconsistent`**Marcador para ignorar**`ignore-inconsistent`

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

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

Esta verificação se aplica a todos os componentes em um projeto que têm *Permitir propagação de tradução* ativada.

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

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

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

Ver também:

Mantendo traduções iguais entre componentes

Letra Kashida usada

Novo na versão 3.5.

Resumo

As letras decorativas de kashida não devem ser usadas

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.KashidaCheck`

Identificador da verificação

`kashida`

Marcador para ignorar

`ignore-kashida`

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

Ver também:

[Kashida na Wikipédia](#)

Links Markdown

Novo na versão 3.5.

Resumo

Os links Markdown não correspondem com a fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.MarkdownLinkCheck`

Identificador da verificação

`md-link`

Marcador para habilitar

`md-text`

Marcador para ignorar

`ignore-md-link`

Os links Markdown não correspondem com a fonte.

Ver também:

[Links Markdown](#)

Referências Markdown

Novo na versão 3.5.

Resumo

As referências de links Markdown não correspondem com a fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.MarkdownRefLinkCheck`

Identificador da verificação

`md-reflink`

Marcador para habilitar

`md-text`

Marcador para ignorar

`ignore-md-reflink`

As referências de link Markdown não correspondem à fonte.

Ver também:

[Links Markdown](#)

Sintaxe Markdown

Novo na versão 3.5.

Resumo

A sintaxe Markdown não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.MarkdownSyntaxCheck`

Identificador da verificação

`md-syntax`

Marcador para habilitar

`md-text`

Marcador para ignorar

`ignore-md-syntax`

A sintaxe Markdown não corresponde à fonte

Ver também:

[Elementos span Markdown](#)

Comprimento máximo da tradução

Resumo

A tradução não deve exceder o comprimento

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.MaxLengthCheck`

Identificador da verificação

`max-length`

Marcador para habilitar

`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 marcador deve ser definido como um par `chave:valor`, como `max-length:100`.

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

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

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

Tamanho máximo da tradução

Resumo

O texto renderizado da tradução não deve exceder o tamanho dado

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.render.MaxSizeCheck`

Identificador da verificação

`max-size`

Marcador para habilitar

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

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

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

Dica: Você pode querer definir as diretivas `font-*` na [Configuração de componente](#) para ter a mesma fonte configurada para todos os textos dentro de um componente. Você pode sobrescrever esses valores por texto, caso você precise personalizá-la por texto.

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

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

Ver também:

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

\n não correspondente

Resumo

O número de `\n` literais na tradução não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EscapedNewlineCountingCheck`

Identificador da verificação

`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 o número de literais `\n` na tradução não corresponder à fonte.

Caractere de dois pontos não correspondente

Resumo

Fonte e tradução não terminam com um caractere de dois pontos

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndColonCheck`

Identificador da verificação

`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).

Ver também:

[Dois pontos na Wikipédia](#)

Reticências não correspondentes

Resumo

A fonte e a tradução não terminam com reticências

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndEllipsisCheck`

Identificador da verificação

`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.

Ver também:

[Reticências na Wikipédia](#)

Ponto de exclamação não correspondente

Resumo

Fonte e tradução não terminam com uma interrogação

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndExclamationCheck`

Identificador da verificação

`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).

Ver também:

[Ponto de exclamação na Wikipédia](#)

Ponto final não correspondente

Resumo

Fonte e tradução não terminam com um ponto final

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndStopCheck`

Identificador da verificação

`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).

Ver também:

[Ponto final na Wikipédia](#)

Ponto de interrogação não correspondente

Resumo

Fonte e tradução não terminam com uma interrogação

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndQuestionCheck`

Identificador da verificação

`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).

Ver também:

[Ponto de interrogação na Wikipédia](#)

Ponto e vírgula não correspondente

Resumo

Fonte e tradução não terminam com ponto e vírgula

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndSemicolonCheck`

Identificador da verificação

`end_semicolon`

Marcador para ignorar

`ignore-end-semicolon`

Verifica se os ponto e vírgulas no final das sentenças são replicados entre a fonte e a tradução.

Ver também:

[Ponto e vírgula na Wikipédia](#)

Quebras de linha descasadas

Resumo

O número de novas linhas na tradução não corresponde à fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.NewLineCountCheck`

Identificador da verificação

`newline-count`

Marcador para ignorar

`ignore-newline-count`

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

Faltam plurais

Resumo

Algumas formas de plural não estão traduzidas

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.consistency.PluralsCheck`

Identificador da verificação

`plurals`

Marcador para ignorar

`ignore-plurals`

Verifica se todas as formas plurais de um texto fonte foram traduzidas. Especificações sobre como cada forma plural é usada podem ser encontradas na definição do texto.

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

Espaços reservados

Novo na versão 3.9.

Resumo

Faltam alguns espaços reservados na tradução

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.placeholders.PlaceholderCheck`

Identificador da verificação

`placeholders`

Marcador para habilitar

`placeholders`

Marcador para ignorar

`ignore-placeholders`

Alterado na versão 4.3: Você pode usar uma expressão regular como espaço reservado.

Alterado na versão 4.13: Com o marcador `case-insensitive`, os espaços reservados não diferenciam maiúsculas de minúsculas.

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

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

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

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

Você também pode ter espaços reservados que não diferenciam maiúsculas de minúsculas:

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

Ver também:

[Personalizando comportamento usando marcadores](#)

Espaçamento de pontuação

Novo na versão 3.9.

Resumo

Faltando espaço rígido antes de sinal de pontuação dupla

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.PunctuationSpacingCheck`

Identificador da verificação

`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.

Ver também:

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

Expressão regular

Novo na versão 3.9.

Resumo

A tradução não corresponde à expressão regular

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.placeholders.RegexCheck`

Identificador da verificação

`regex`

Marcador para habilitar

`regex`

Marcador para ignorar

`ignore-regex`

A tradução não corresponde à expressão regular. A expressão é extraída do arquivo de tradução ou definida manualmente usando o marcador `regex`:

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

Mesmos plurais

Resumo

Algumas formas de plural estão traduzidas da mesma forma

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.consistency.SamePluralsCheck`

Identificador da verificação

`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

Fonte e tradução não iniciam com uma nova linha

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.BeginNewlineCheck`

Identificador da verificação

`begin_newline`

Marcador para ignorar

`ignore-begin-newline`

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

Ver também:

[*Linha em branco no final*](#)

Espaços no início

Resumo

Fonte e tradução não iniciam com o mesmo número de espaços

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.BeginSpaceCheck`

Identificador da verificação

`begin_space`

Marcador para ignorar

`ignore-begin-space`

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

Linha em branco no final

Resumo

Fonte e tradução não terminam com uma nova linha

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndNewlineCheck`

Identificador da verificação

`end_newline`

Marcador para ignorar

`ignore-end-newline`

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

Ver também:

[*Nova linha no início*](#)

Espaço no final

Resumo

Fonte e tradução não terminam com um espaço

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.EndSpaceCheck`

Identificador da verificação

`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 não alterada

Resumo

A fonte e a tradução são idênticas

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.same.SameCheck`

Identificador da verificação

`same`

Marcador para ignorar

`ignore-same`

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

Esta verificação pode ajudar a encontrar textos erroneamente não traduzidos.

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

Ver também:

Configuração de componente, Personalizando comportamento usando marcadores

HTML inseguro

Novo na versão 3.9.

Resumo

A tradução usa marcação de HTML inseguro

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.SafeHTMLCheck`

Identificador da verificação

`safe-html`

Marcador para habilitar

`safe-html`

Marcador para ignorar

`ignore-safe-html`

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

Dica: Quando o marcador `md-text` também é usado, os links de estilo Markdown também são permitidos.

Ver também:

The HTML check is performed by the [Ammonia](#) library.

URL

Novo na versão 3.5.

Resumo

A tradução não contém uma URL

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.URLCheck`

Identificador da verificação

`url`

Marcador para habilitar

`url`

Marcador para ignorar

`ignore-url`

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

Marcação XML

Resumo

As tags XML na tradução não correspondem com a fonte

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.XMLTagsCheck`

Identificador da verificação

`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 desabilitada 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 um XML válido

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.markup.XMLValidityCheck`

Identificador da verificação

`xml-invalid`

Marcador para ignorar

`ignore-xml-invalid`

A marcação XML não é válida.

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

Espaço com largura zero

Resumo

A tradução contém um caractere extra de espaço com largura zero

Escopo

textos traduzidos

Classe da verificação

`weblate.checks.chars.ZeroWidthSpaceCheck`

Identificador da verificação

`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.

Ver também:

[Espaço de largura zero na Wikipédia](#)

1.6.4 Verificações de texto fonte

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

Reticências

Resumo

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

Escopo

textos fonte

Classe da verificação

`weblate.checks.source.EllipsisCheck`

Identificador da verificação

`ellipsis`

Marcador para ignorar

`ignore-ellipsis`

Isto falha quando o texto usa três pontos (. . .) quando deveria usar um caractere de reticências (...).

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

Ver também:

[Reticências na Wikipédia](#)

Sintaxe de MessageFormat do ICU

Novo na versão 4.9.

Resumo

Erros de sintaxe em textos de MessageFormat do ICU.

Escopo

textos fonte

Classe da verificação

`weblate.checks.icu.ICUSourceCheck`

Identificador da verificação

`icu_message_format_syntax`

Marcador para habilitar

`icu-message-format`

Marcador para ignorar

`ignore-icu-message-format`

Ver também:

MessageFormat do ICU

Não traduzido há muito tempo

Novo na versão 4.1.

Resumo

Este texto não foi traduzido há muito tempo

Escopo

textos fonte

Classe da verificação

`weblate.checks.source.LongUntranslatedCheck`

Identificador da verificação

`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, tornando difícil a tradução.

Várias verificações com falha

Resumo

As traduções em diversos idiomas têm verificações com falha

Escopo

textos fonte

Classe da verificação

`weblate.checks.source.MultipleFailingCheck`

Identificador da verificação

`multiple_failures`

Marcador para ignorar

`ignore-multiple-failures`

Várias traduções deste texto têm falhas nas verificações de qualidade. Isto é normalmente uma indicação de que algo poderia ser feito para melhorar o texto fonte.

Esta falha na verificação pode muitas vezes ser causada por um ponto final de uma frase, ou por questões menores similares que os tradutores tendem a corrigir na tradução, enquanto que seria melhor corrigi-la no texto fonte.

Várias variáveis sem nome

Novo na versão 4.1.

Resumo

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

Escopo

textos fonte

Classe da verificação

`weblate.checks.format.MultipleUnnamedFormatsCheck`

Identificador da verificação

`unnamed_format`

Marcador para ignorar

`ignore-unnamed-format`

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

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

Não pluralizado

Resumo

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

Escopo

textos fonte

Classe da verificação

`weblate.checks.source.OptionalPluralCheck`

Identificador da verificação

`optional_plural`

Marcador para ignorar

`ignore-optional-plural`

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

Por exemplo, com o Gettext em Python poderia ser:

```
from gettext import ngettext

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

1.7 Pesquisando

Novo na versão 3.9.

Consultas avançadas usando operações booleanas, parênteses ou pesquisas específicas de campo podem ser usadas para encontrar os textos que você deseja.

Quando nenhum campo é definido, a pesquisa acontece em textos fonte, destino e de contexto.

The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo and links to Dashboard, Projects, Languages, and Checks. Below this is a light-colored header with 'Watched translations 0', 'Suggested translations 0', 'Insights', and a 'Search' button. The main content area is titled 'Search' and contains a search bar with a dropdown menu set to 'All strings'. Below the search bar is an 'Advanced query builder' section with a 'Search for...' input, an 'Exact' checkbox, and an 'Add' button. There is also a 'Strings with suggestions' dropdown and an 'Add' button. Below this is a 'String changed after' section with a date input 'mm/dd/yyyy' and an 'Add' button. The 'Query examples' section lists several queries with their corresponding syntax and an 'Add' button for each:

Query examples	Syntax	Action
Review strings changed by other users	<code>changed:>=2023-01-29 AND NOT changed_by:testuser</code>	Add
Translated strings	<code>state:>=translated</code>	Add
Strings with comments	<code>has:comment</code>	Add
Strings with any failing checks	<code>has:check</code>	Add
Strings with suggestions from others	<code>has:suggestion AND NOT suggestion_author:testuser</code>	Add
Approved strings with suggestions	<code>state:approved AND has:suggestion</code>	Add
All untranslated strings added the past month	<code>added:>=2023-01-29 AND state:<=needs-editing</code>	Add
Translated strings in a certain language	<code>is:translated AND language:cs</code>	Add

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

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

1.7.1 Pesquisa simples

Qualquer frase digitada na caixa de pesquisa é dividida em palavras. Os textos que contêm qualquer uma delas são mostradas. Para procurar uma frase exata, coloque “a frase de pesquisa” entre aspas (ambas aspas simples (') e duplas (")) funcionarão: "este é um texto entre aspas" ou 'outro texto entre aspas'.

1.7.2 Campos

source:TEXTO

Pesquisa em texto fonte sem diferenciar maiúscula de minúscula.

target:TEXTO

Pesquisa em texto destino sem diferenciar maiúscula de minúscula.

context:TEXTO

Pesquisa em texto de contexto sem diferenciar maiúscula de minúscula.

key:TEXTO

Pesquisa em texto chave sem diferenciar maiúscula de minúscula.

note:TEXTO

Pesquisa em descrição de texto fonte sem diferenciar maiúscula de minúscula.

location:TEXTO

Pesquisa em texto de local sem diferenciar maiúscula de minúscula.

priority:NÚMERO

Prioridade do texto.

id:NUMBER

String unique identifier.

added:DATA E HORA

Data e hora de quando o texto foi adicionado ao Weblate.

state:TEXTO

Pesquisa por estados do texto (approved, translated, needs-editing, empty, read-only), tem suporte a *Operações de campo*.

pending:BOOLEANO

Texto pendente para descarga para o VCS.

has:TEXTO

Pesquisa por texto com atributos - plural, context, suggestion, comment, check, dismissed-check, translation, variant, screenshot, flags, explanation, glossary, note, label.

is:TEXTO

Pesquisa por traduções pendentes (pending). Também pode procurar por todos os estados de texto (approved, translated, untranslated, needs-editing, read-only).

language:TEXTO

Idioma de destino do texto.

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

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

changed:DATA E HORA

O conteúdo do texto foi alterado na data, tem suporte a *Operações de campo*.

change_time:DATA E HORA

Texto foi alterado na data, tem suporte a *Operações de campo*; ao contrário de `changed`, isso inclui eventos que não alteram o conteúdo e você pode aplicar uma filtragem de ação personalizada utilizando `change_action`.

change_action:TEXTO

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

check:TEXTO

O texto falhou na verificação, consulte *Verificações e correções* para identificadores de verificação.

dismissed_check:TEXTO

O texto dispensou a verificação, consulte *Verificações e correções* para identificadores de verificação.

comment:TEXTO

Pesquisa nos comentários de usuários.

resolved_comment:TEXTO

Pesquisa nos comentários resolvidos.

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:TEXTO

Pesquisa em etiquetas.

screenshot:TEXTO

Pesquisa nas capturas de tela.

1.7.3 Operadores booleanos

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

1.7.4 Operações de campo

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

state:>=translated

Estado é `translated` ou melhor (`approved`).

changed:2019

Alterado no ano de 2019.

changed:[2019-03-01 to 2019-04-01]

Alterado entre as duas datas dadas.

1.7.5 Operadores exatos

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

1.7.6 Pesquisando por alterações

Novo na versão 4.4.

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

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

1.7.7 Expressões regulares

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

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

1.7.8 Consultas predefinidas

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

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation for the string '%(count)s word'. The interface includes a sidebar with project details, a main translation area with a custom search filter, and a right sidebar with glossary and string information.

Translation Area:

- English:** Singular: '%(count)s word', Plural: '%(count)s words'
- Czech, One:** '%(count)s slovo'
- Czech, Few:** '%(count)s slova'
- Czech, Many:** '%(count)s slov'

Custom search: '%(count)s word'

String information:

- Screenshot context:** No screenshot currently associated.
- 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

Comments:

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

Scope: Translation comment, discussions with other translators

New comment:

You can use Markdown and mention users by @username.

Buttons: Save and continue, Save and stay, Suggest, Skip

1.7.9 Ordenando os resultados

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

The screenshot displays the Weblate web interface for a project named 'Django' in the 'Czech' language. The main area shows a translation form for a string: 'The string uses three dots (...) instead of an ellipsis character (...)'. The form includes fields for 'English' and 'Czech' translations, a 'Needs editing' checkbox, and buttons for 'Save and continue', 'Save and stay', 'Suggest', and 'Skip'. A dropdown menu is open over the 'Position and priority' button, listing various metadata fields. The right sidebar contains a 'Glossary' section and a 'String information' section with details like 'Screenshot context', 'Explanation', 'Labels', 'Flags', 'Source string location', 'String age', 'Source string age', and 'Translation file'. The bottom section is for 'New comment' with a text area and a 'Save' button.

Powered by Weblate 4.16 · [About Weblate](#) · [Legal](#) · [Contact](#) · [Documentation](#) · [Donate to Weblate](#)

1.8 Fluxos de trabalho de tradução

Usar o Weblate é um processo que aproxima seus usuários de você, aproximando você de seus tradutores. Depende de você decidir quantos recursos deseja usar.

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

1.8.1 Acesso à tradução

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

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

1.8.2 Estados de tradução

Cada texto traduzida pode estar em um dos seguintes estados:

Não traduzido

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

Necessita edição

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

Aguardando revisão

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

Aprovado

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

Este estado está disponível apenas quando revisões estão habilitadas.

Sugestões

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

Os estados são representados nos arquivos de tradução quando possível.

Dica: No caso do formato de arquivo que você utiliza não ter suporte ao armazenamento de estados, você pode querer utilizar a extensão *Marcar traduções não alteradas como "Necessita edição"* para marcar textos inalterados como precisando de edição.

Ver também:

Capacidades dos tipos de tradução, Fluxos de trabalho de tradução

1.8.3 Tradução direta

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

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

Configuração	Valor	Nota
Habilitar revisões	desativada	Configurada a nível de projeto.
Habilitar sugestões	ativada	É útil para os usuários serem capazes de sugerir quando não têm certeza.
Votação de sugestões	desativada	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	<i>Usuários</i>	Ou Traduzir com <i>controle de acesso por projeto</i> .
Grupo de revisores	N/D	Não usada.

1.8.4 Revisão por pares

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

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

Configuração	Valor	Nota
Habilitar revisões	desativada	Configurada a nível de projeto.
Habilitar sugestões	ativada	
Votação de sugestões	desativada	
Aceitar sugestões automaticamente	1	Você pode definir um valor mais alto para exibir mais revisões por pares.
Grupo de tradutores	<i>Usuários</i>	Ou <i>Traduzir</i> com <i>controle de acesso por projeto</i> .
Grupo de revisores	N/D	Não usada, todos os tradutores revisam.

1.8.5 Revisores dedicados

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

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

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

Configuração	Valor	Nota
Habilitar revisões	ativada	Configurada a nível de projeto.
Habilitar sugestões	desativada	É útil para os usuários serem capazes de sugerir quando não têm certeza.
Votação de sugestões	desativada	
Aceitar sugestões automaticamente	0	
Grupo de tradutores	<i>Usuários</i>	Ou <i>Traduzir</i> com <i>controle de acesso por projeto</i> .
Grupo de revisores	<i>Revisores</i>	Ou <i>Revisar</i> com <i>controle de acesso por projeto</i> .

1.8.6 Ativando revisões

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

Webblate Dashboard Projects Languages Checks

WebblateOrg / Settings

Basic Access **Workflow** Components

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

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

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

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

Language aliases ⓘ

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

☐ Enable reviews ⓘ
Requires dedicated reviewers to approve translations.

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

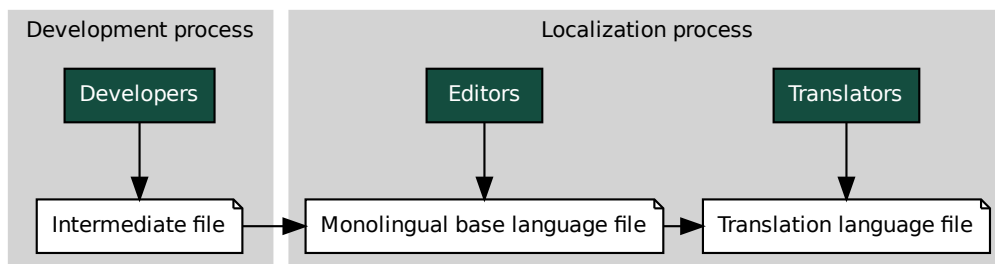
Save

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

1.8.7 Rota de qualidade para os textos fonte

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

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

**Ver também:**

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

1.8.8 Revisões de textos fonte

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

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

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

Ver também:

Formatos bilíngues e monolíngues, Revisores dedicados, labels, Comentários

1.9 Perguntas frequentes

1.9.1 Configuração

Como criar um fluxo de trabalho automatizado?

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

1. Configure seu repositório Git para informar Weblate quando houver alguma alteração, veja *Ganchos de notificação* para informações sobre como fazer isso.
2. Defina uma URL de push na sua *Configuração de componente* no Weblate, isto permite que o Weblate faça o push das alterações para o seu repositório.
3. Ative o *Push ao fazer commit* 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.

Ver também:

Localização contínua, Evitando conflitos de mesclagem

Como acessar os repositórios por SSH?

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

Como resolver conflitos de mesclagem nas traduções?

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

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

Nota: 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 usuário 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

# Rebase changes (if Weblate is configured to do rebases)
git rebase origin/main

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

# Open Weblate for translation:
wlc unlock
```

Se você estiver usando vários ramos no Weblate, você pode fazer o mesmo com todos eles:

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

# Merge QA_4_7 branch:
```

(continua na próxima página)

(continuação da página anterior)

```
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 arquivos PO gettext, há uma forma de mesclar conflitos de uma forma semiautomática:

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

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

# Update Weblate remote:
git remote update weblate

# Merge Weblate changes:
git merge weblate/main

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

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

Ver também:

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

Como fazer para traduzir vários ramos ao mesmo tempo?

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

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

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

Ver também:*Mantendo traduções iguais entre componentes***Como traduzir projetos multiplataforma?**

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

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

Ver também:*Mantendo traduções iguais entre componentes***Como exportar o repositório Git que o Weblate usa?**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Como posso verificar se o meu Weblate está configurado corretamente?

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

Ver também:

Monitorando o Weblate, Monitorando status 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_COMMITTER_EMAIL` e `DEFAULT_COMMITTER_NAME`.

O autor de cada commit (se o VCS subjacente o suportar) ainda é registrado corretamente como o usuário 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 usuário anônimo (veja `ANONYMOUS_USER_NAME`). Você pode alterar o nome e o e-mail na interface de gerenciamento.

Ver também:

Configuração de componente

Como mover arquivos no repositório sem perder histórico no Weblate?

Para manter o histórico, comentários ou capturas de tela vinculados aos textos após alterar a localização dos arquivos, você precisa garantir que esses textos nunca sejam excluídos do Weblate. Essas remoções podem acontecer caso o repositório Weblate seja atualizado, mas a configuração do componente ainda aponta para os arquivos antigos. Isso faz com que o Weblate presuma que deveria excluir todas as traduções.

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

1. Bloqueie o componente afetado no Weblate.
2. Faça commit de todas as alterações pendentes e mescle-as no repositório upstream.
3. Desabilite o recebimento de webhooks *Configuração de projeto*; isso evita que o Weblate veja imediatamente as mudanças no repositório.
4. Faça todas as alterações necessárias no repositório (por exemplo, usando `git mv`), faça o push delas para o repositório upstream.
5. Altere o *Configuração de componente* para corresponder à nova configuração; ao alterar a configuração, o Weblate buscará o repositório atualizado e observará os locais alterados, mantendo os textos existentes.
6. Desbloqueie o componente e reabilite 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*.
- Você pode se inscrever para quaisquer alterações feitas em *Notificações* e depois verificar outras contribuições à medida que elas chegam por e-mail.
- Há uma ferramenta de revisão disponível na parte inferior da vista de tradução, onde você pode optar por navegar pelas traduções feitas por outros desde uma data determinada.

Ver também:

Fluxos de trabalho de tradução

Como posso fornecer feedback sobre um texto fonte?

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

Ver também:

report-source, *Comentários*

Como posso usar as traduções existentes enquanto traduzo?

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

Ver também:

Configurando sugestões automáticas, Sugestões automáticas, Memória de tradução

O Weblate atualiza arquivos de tradução além de traduções?

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

Ver também:

updating-target-files

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

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

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

Ver também:

Definições de idioma

Weblate pode destacar alterações em um texto aproximado?

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

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

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

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

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

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

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

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

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

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

Ver também:

updating-target-files

1.9.3 Soluções de problemas

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

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

A maneira mais fácil de fazer isto é executar:

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

Ver também:

`DATA_DIR`

Ao acessar ao site recebo um erro de “Bad Request (400)”

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

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

Ver também:

Configuração de hosts permitidos

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

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

- Caso a tradução para o idioma fonte seja desejada, altere o *Idioma fonte* nas configurações do projeto. Você pode querer usar *English (Developer)* como idioma fonte, ou usar *Rota de qualidade para os textos fonte*.
- Caso o arquivo de tradução para o idioma fonte não seja necessário, remova-o do repositório.
- Caso o arquivo de tradução para o idioma fonte seja necessário, mas deva ser ignorado pelo Weblate, ajuste o *Filtro de idioma* para excluí-lo.

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

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

1.9.4 Recursos

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

Weblate atualmente não tem suporte nativo para nada além de *Git* (com suporte estendido para *Pull requests do GitHub*, *Gerrit* e *Subversion*) e *Mercurial*, mas é possível escrever backends para outros VCSs.

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

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

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

Ver também:

Integração com controle de versão

Como o Weblate credita os tradutores?

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

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

Ver também:

`list_translators`, `../devel/reporting`

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

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

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

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

Estes são códigos de idioma definidos por [RFC 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.

Ver também:

[Definições de idioma](#), [Estilo de código de idioma](#), [Adicionando novas traduções](#)

1.10 Formatos de arquivos suportados

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

Ver também:

[Translation Related File Formats](#)

Nota: Ao escolher um formato de arquivo para o seu aplicativo, é melhor colocar algum formato bem estabelecido no kit de ferramentas/plataforma que você usa. Dessa forma, seus tradutores podem usar adicionalmente todas as ferramentas que estão acostumados, e provavelmente contribuirão para o seu projeto.

1.10.1 Formatos bilíngues e monolíngues

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

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

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

1.10.2 Detecção automática

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

1.10.3 Capacidades dos tipos de tradução

Capacidades de todos os formatos suportados:

Formato	Linguali- dade <small>Página 79,</small>	Plu- rais <small>Página 79, 2</small>	Descri- ção <small>Página 79, 3</small>	Con- texto <small>Página 79,</small>	Lo- cal <small>Página 79, 5</small>	Marcado- res <small>Página 79, 8</small>	Estados adicio- nais <small>Página 79, 6</small>
<i>GNU gettext</i>	bilíngue	sim	sim	sim	sim	sim ⁹	necessita edição
<i>Gettext mo- nolíngue</i>	mono	sim	sim	sim	sim	sim ⁹	necessita edição
<i>XLIFF</i>	ambos	sim	sim	sim	sim	sim ¹⁰	necessita edição, aprovado
<i>Propriedades Java</i>	ambos	não	sim	não	não	não	
<i>Arquivos lang mi18n</i>	mono	não	sim	não	não	não	
<i>Propriedades GWT</i>	mono	sim	sim	não	não	não	
<i>Tradu- ções para Joomla</i>	mono	não	sim	não	sim	não	
<i>.ts de Qt Linguist</i>	ambos	sim	sim	não	sim	sim <small>Página 79, 10</small>	necessita edição
<i>Recurso de textos de Android</i>	mono	sim	sim ⁷	não	não	sim <small>Página 79, 10</small>	
<i>Textos de Apple iOS</i>	ambos	não	sim	não	não	não	
<i>Textos do PHP</i>	mono	não ¹¹	sim	não	não	não	

continua na próxima página

Tabela 1 – continuação da página anterior

Formato	Linguali- dade ^{Página 79,}	Plurais ²	Descri- ção ³	Contexto ⁴	Local ⁵	Marcado- res ⁸	Estados adicio- nais ⁶
<i>Arquivos JSON</i>	mono	não	não	não	não	não	
<i>Arquivos i18next de JSON</i>	mono	sim	não	não	não	não	
<i>Arquivos go-i18n de JSON</i>	mono	sim	não	não	não	não	
<i>arquivos gotext JSON</i>	mono	sim	sim	não	sim	não	
<i>Arquivo ARB</i>	mono	sim	sim	não	não	não	
<i>JSON para WebExtension</i>	mono	sim	sim	não	não	não	
<i>Arquivos de recursos .XML</i>	mono	não	sim	não	não	sim ^{Página 79, 1C}	
<i>Arquivos de Resource-Dictionary</i>	mono	não	não	não	não	sim ^{Página 79, 1C}	
<i>Arquivos CSV</i>	ambos	não	sim	sim	sim	não	necessita edição
<i>Arquivos YAML</i>	mono	não	sim	não	não	não	
<i>Arquivos Ruby YAML</i>	mono	sim	sim	não	não	não	
<i>Arquivos DTD</i>	mono	não	não	não	não	não	
<i>Arquivos XML simples</i>	mono	não	não	não	não	sim ^{Página 79, 1C}	
<i>Arquivos RC do Windows</i>	mono	não	sim	não	não	não	
<i>Excel Open XML</i>	mono	não	sim	sim	sim	não	necessita edição
<i>Arquivos de metadados de loja de aplicativos</i>	mono	não	não	não	não	não	
<i>Arquivos de legenda</i>	mono	não	não	não	sim	não	
<i>Arquivos HTML</i>	mono	não	não	não	não	não	
<i>Formato OpenDocument</i>	mono	não	não	não	não	não	

continua na próxima página

Tabela 1 – continuação da página anterior

Formato	Linguali- dade ^{Página 79,}	Plurais ²	Descri- ção ³	Contexto ⁴	Local ⁵	Marcado- res ⁸	Estados adicio- nais ⁶
<i>Formato IDML</i>	mono	não	não	não	não	não	
<i>Traduções de INI</i>	mono	não	não	não	não	não	
<i>Traduções de Inno Setup INI</i>	mono	não	não	não	não	não	
<i>Formato TermBase eXchange</i>	bilíngue	não	sim	não	não	sim ^{Página 79, 10}	
<i>Arquivos texto</i>	mono	não	não	não	não	não	
<i>Formato stringsdict</i>	mono	sim	sim	não	não	não	
<i>Formato Fluent</i>	mono	não ¹²	sim	não	não	não	

Textos somente leitura

Novo na versão 3.10.

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

1.10.4 GNU gettext

O formato mais usado para traduzir software livre.

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

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

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

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

(continua na próxima página)

¹ Veja [Formatos bilíngues e monolíngues](#)

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

³ As descrições de textos fonte podem ser usadas para passar informações adicionais sobre o texto para traduzir.

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

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

⁸ Veja [Personalizando comportamento usando marcadores](#)

⁶ Estados adicionais suportados pelo formato do arquivo, além de “Não traduzido” e “Traduzido”.

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 [Especificando marcadores de tradução](#).

⁷ Comentário XML colocado antes do elemento `<string>`, analisado como uma descrição do texto fonte.

¹¹ Os plurais são suportados apenas para Laravel, que usa em sintaxe de textos para defini-los, veja [Localização em Laravel](#).

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

(continuação da página anterior)

```
msgstr "Úterý"

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

Configuração de componente típica do Weblate

Máscara de arquivo	po/* .po
Arquivo de idioma da base monolíngue	Vazio
Modelo para novas traduções	po/messages.pot
Formato de arquivo	Arquivo Gettext PO

Ver também:

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

Gettext monolíngue

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

O arquivo gettext PO monolíngue normalmente se parece com este:

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

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

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

Enquanto o arquivo de idioma base será:

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

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

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

Configuração de componente típica do Weblate

Máscara de arquivo	po/* .po
Arquivo de idioma da base monolíngue	po/en .po
Modelo para novas traduções	po/messages .pot
Formato de arquivo	Arquivo Gettext PO (monolíngue)

1.10.5 XLIFF

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

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

O Weblate tem suporte a XLIFF em várias variantes:

Arquivo de tradução XLIFF

Arquivo XLIFF simples onde o conteúdo dos elementos é armazenado como texto simples (todos os elementos XML sendo escapados).

XLIFF com suporte transponível

XLIFF padrão que oferece suporte a objetos colocáveis e outros elementos XML.

XLIFF com extensões gettext

XLIFF enriquecido pelo [XLIFF 1.2 Representation Guide for Gettext PO](#) para oferecer suporte a plurais.

Ver também:

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

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

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

- O atributo `state` só é adicionado no caso de texto ser marcado como necessitando de edição.
- O atributo `approved` só é adicionado no caso de texto ter sido revisto.
- Em outros casos os atributos não são adicionados, mas são atualizados no caso de estarem presentes.

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

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

Ver também:

[Revisores dedicados](#)

Espaços em branco e novas linhas em XLIFF

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

Por exemplo:

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

Especificando marcadores de tradução

Você pode especificar marcadores de tradução adicionais (veja *Personalizando comportamento usando marcadores*) utilizando o atributo `weblate-flags`. Weblate também entende os atributos `maxwidth` e `font` da especificação XLIFF:

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

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

Chaves de textos

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

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

Configuração de componente típica do Weblate para XLIFF bilíngue

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

Configuração de componente típica do Weblate para XLIFF monolíngue

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

Ver também:

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

1.10.6 Propriedades Java

Formato nativo do Java para traduções.

As propriedades Java são geralmente usadas como traduções monolíngues.

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

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

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

Ver também:

[Java properties na Wikipédia](#), [Mozilla and Java properties files](#), [Arquivos lang mi18n](#), [Propriedades GWT](#), [updating-target-files](#), [Formatar o arquivo de propriedades Java](#), [Limpar arquivos de tradução](#)

1.10.7 Arquivos lang mi18n

Novo na versão 4.7.

Formato de arquivo usado para localização de JavaScript por [mi18n](#). Sintaticamente, ele corresponde a [Propriedades Java](#).

<i>Configuração de componente</i> típica do Weblate	
Máscara de arquivo	<code>*.lang</code>
Arquivo de idioma da base monolíngue	<code>en-US.lang</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de arquivo	<i>Arquivo lang mi18n</i>

Ver também:

[mi18n](#), [Mozilla and Java properties files](#), [Propriedades Java](#), [updating-target-files](#), [Formatar o arquivo de propriedades Java](#), [Limpar arquivos de tradução](#)

1.10.8 Propriedades GWT

Formato nativo de GWT para traduções.

As propriedades GWT são geralmente usadas como traduções monolíngues.

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

Ver também:

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

1.10.9 Traduções de INI

Novo na versão 4.1.

Formato de arquivo INI para traduções.

As traduções de INI são geralmente usadas como traduções monolíngues.

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

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

Ver também:

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

1.10.10 Traduções de Inno Setup INI

Novo na versão 4.1.

Formato de arquivo Inno Setup INI para traduções.

As traduções de Inno Setup INI são geralmente usadas como traduções monolíngues.

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

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

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

Ver também:

[INI Files](#), [Traduções para Joomla](#), [Traduções de INI](#)

1.10.11 Traduções para Joomla

Novo na versão 2.12.

Formato nativo de Joomla para traduções.

As traduções de Joomla são geralmente usadas como traduções monolíngues.

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

Ver também:

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

1.10.12 .ts de Qt Linguist

Formato de tradução usado em aplicativos baseados em Qt.

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

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

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

Ver também:

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

1.10.13 Recurso de textos de Android

Formato de arquivo específico do Android para traduzir aplicativos.

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

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

Ver também:

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

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

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

se torna:

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

O *string-array* que aponta para os elementos *string* deve ser armazenado em um arquivo diferente e não ser disponibilizado para tradução.

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

Dica: Para evitar a tradução de alguns textos, eles podem ser marcados como não traduzíveis. Isso pode ser especialmente útil para referências de textos:

```
<string name="foobar" translatable="false">@string/foo</string>
```

1.10.14 Textos de Apple iOS

Formato de arquivo normalmente usado para traduzir aplicativos do Apple iOS, mas também padronizado pelo PWG 5100.13 e usado no NeXTSTEP/OpenSTEP.

Os textos de Apple iOS geralmente são usados como monolingue.

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

Ver também:

Formato stringsdict, Documentação de “arquivos de textos” da Apple, Message Catalog File Format no PWG 5100.13, Mac OSX strings

1.10.15 Textos do PHP

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

Exemplo de arquivo:

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

Configuração de componente típica do Weblate

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

Textos do PHP de Laravel

Alterado na versão 4.1.

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

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

Ver também:

[PHP, Localização em Laravel](#)

1.10.16 Arquivos JSON

Novo na versão 2.0.

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

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

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

O Weblate atualmente tem suporte a várias variantes das traduções JSON:

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

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

Exemplo de arquivo:

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

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

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

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

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

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

Configuração de componente típica do Weblate

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

Ver também:

[JSON](#), [updating-target-files](#), [Personalizar saída JSON](#), [Limpar arquivos de tradução](#),

1.10.17 Arquivos i18next de JSON

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

Alterado na versão 4.15.1: O suporte para a variante v4 deste formato foi adicionado.

Dica: In case you use plurals, it is recommended to use v4 as that aligned plural handling with CLDR. Older versions have different plural rules for some languages which are not correct.

i18next é uma estrutura de internacionalização escrita dentro e para JavaScript. O Weblate tem suporte a seus arquivos de localização com recursos como plurais.

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

Nota: O Weblate suporta as variantes i18next JSON v3 e v4. Escolha o formato de arquivo correto que corresponda ao seu ambiente.

As variantes v2 e v1 são em sua maioria compatíveis com v3, com exceção de como os plurais são tratados.

Exemplo de arquivo:

```
{
  "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 arquivo	langs/*.json
Arquivo de idioma da base monolíngue	langs/en.json
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo JSON i18next v3

Ver também:

[JSON](#), [JSON Format de i18next](#), [updating-target-files](#), [Personalizar saída JSON](#), [Limpar arquivos de tradução](#)

1.10.18 Arquivos go-i18n de JSON

Novo na versão 4.1.

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

Alterado na versão 4.16: Weblate supports both v1 and v2 formats, please choose corresponding version when selecting file format.

Configuração de componente típica do Weblate

Máscara de arquivo	langs/*.json
Arquivo de idioma da base monolíngue	langs/en.json
Modelo para novas traduções	Vazio
Formato de arquivo	go-i18n v1 JSON file

Ver também:

[JSON](#), [go-i18n](#), [updating-target-files](#), [Personalizar saída JSON](#), [Limpar arquivos de tradução](#),

1.10.19 arquivos gotext JSON

Novo na versão 4.15.1.

As traduções do gotext são monolíngues, por isso é recomendável especificar um arquivo base com (o que é mais comum) strings em inglês.

<i>Configuração de componente</i> típica do Weblate	
Máscara de arquivo	<code>internal/translations/locales/*/messages.gotext.json</code>
Arquivo de idioma da base monolíngue	<code>internal/translations/locales/en-GB/messages.gotext.json</code>
Modelo para novas traduções	<i>Vazio</i>
Formato de arquivo	<code>`arquivo JSON gotext`</code>

Ver também:

[JSON](#), [I18n in Go: Gerenciando Traduções](#), [updating-target-files](#), [Personalizar saída JSON](#), [Limpar arquivos de tradução](#),

1.10.20 Arquivo ARB

Novo na versão 4.1.

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

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

Ver também:

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

1.10.21 JSON para WebExtension

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

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

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

Exemplo de arquivo:

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

(continua na próxima página)

(continuação da página anterior)

```

    "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 arquivo	_locales/*/messages.json
Arquivo de idioma da base monolíngue	_locales/en/messages.json
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo JSON para WebExtension

Ver também:

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

1.10.22 Arquivos de recursos .XML

Novo na versão 2.3.

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

Configuração de componente típica do Weblate

Máscara de arquivo	Resources/Language.*.resx
Arquivo de idioma da base monolíngue	Resources/Language.resx
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo de recurso .NET

Ver também:

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

1.10.23 Arquivos de ResourceDictionary

Novo na versão 4.13.

ResourceDictionary é um formato de arquivo XML monolíngue usado para empacotar recursos de textos localizáveis para aplicativos Windows Presentation Foundation (WPF).

<i>Configuração de componente</i> típica do Weblate	
Máscara de arquivo	Languages/* .xaml
Arquivo de idioma da base monolíngue	Language/en .xaml
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo ResourceDictionary

Ver também:

Flat XML, *Arquivos XML simples*, updating-target-files, *Limpar arquivos de tradução*

1.10.24 Arquivos CSV

Novo na versão 2.4.

Os arquivos CSV podem conter uma simples lista de fonte e tradução. O Weblate tem suporte aos seguintes arquivos:

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

Dica: Por padrão, o formato CSV detecta automaticamente a codificação do arquivo. Isso pode não ser confiável em alguns casos de canto e causa penalidade de desempenho. Por favor, escolha a variante de formato de arquivo com codificação para evitar isso (por exemplo *Arquivo CSV (UTF-8)*).

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

Exemplo de arquivo:

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

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

Configuração de componente típica do Weblate para CSV monolíngue

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

Arquivo CSV com vários valores

Novo na versão 4.13.

Esta variante dos arquivos CSV permite armazenar várias traduções por texto.

Ver também:

[CSV](#)

1.10.25 Arquivos YAML

Novo na versão 2.9.

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

Exemplo de um arquivo YAML:

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

Configuração de componente típica do Weblate

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

Ver também:

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

1.10.26 Arquivos Ruby YAML

Novo na versão 2.9.

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

Exemplo de arquivo YAML de i18n do Ruby:

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

Configuração de componente típica do Weblate

Máscara de arquivo	translations/messages.*.yaml
Arquivo de idioma da base monolíngue	translations/messages.en.yaml
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo Ruby YAML

Ver também:[YAML](#), [Arquivos YAML](#)

1.10.27 Arquivos DTD

Novo na versão 2.18.

Exemplo de arquivo DTD:

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

Configuração de componente típica do Weblate

Máscara de arquivo	locale/*.dtd
Arquivo de idioma da base monolíngue	locale/en.dtd
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo DTD

Ver também:[Mozilla DTD format](#)

1.10.28 Arquivos XML simples

Novo na versão 3.9.

Exemplo de um arquivo XML simples:

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

Configuração de componente típica do Weblate

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

Ver também:[Flat XML](#)

1.10.29 Arquivos RC do Windows

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

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

Exemplo de arquivo RC do Windows:

```
LANGUAGE LANG_CZECH, SUBLANG_DEFAULT

STRINGTABLE
BEGIN
    IDS_MSG1                "Hello, world!\n"
    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 arquivo	lang/*.rc
Arquivo de idioma da base monolíngue	lang/en-US.rc
Modelo para novas traduções	lang/en-US.rc
Formato de arquivo	<i>Arquivo RC</i>

Ver também:

Windows RC files

1.10.30 Arquivos de metadados de loja de aplicativos

Novo na versão 3.5.

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

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

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

Configuração de componente típica do Weblate

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

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

1.10.31 Arquivos de legenda

Novo na versão 3.7.

Weblate pode traduzir vários arquivos de legenda:

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

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

Ver também:

[Subtitles](#)

1.10.32 Excel Open XML

Novo na versão 3.2.

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

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

1.10.33 Arquivos HTML

Novo na versão 4.1.

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

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

Ver também:

[HTML](#)

1.10.34 Arquivos 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 arquivos texto simples e oferecido para a tradução. Cada parágrafo é traduzido como um texto separado.

Há três sabores desse formato:

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

Ver também:

[Simple Text Documents](#)

1.10.35 Formato OpenDocument

Novo na versão 4.1.

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

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

Ver também:

[OpenDocument Format](#)

1.10.36 Formato IDML

Novo na versão 4.1.

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

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

1.10.37 Formato TermBase eXchange

Novo na versão 4.5.

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

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

Ver também:

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

1.10.38 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.

<i>Configuração de componente</i> típica do Weblate	
Máscara de arquivo	Resources/*.lproj/Localizable.stringsdict
Arquivo de idioma da base monolíngue	Resources/en.lproj/Localizable.stringsdict ou Resources/Base.lproj/Localizable.stringsdict
Modelo para novas traduções	Vazio
Formato de arquivo	Arquivo Stringsdict

Ver também:

ref:apple, [Formato de arquivo Stringsdict](#)

1.10.39 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 em um idioma pode ser mapeado para uma tradução multivariante complexa em outro idioma.

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

Ver também:

[Site do projeto Fluent](#)

1.10.40 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](#).

Ver também:

[Translation Related File Formats](#)

1.11 Integração com controle de versão

Weblate currently supports *Git* (with extended support for *Pull requests do GitHub*, *Merge requests do GitLab*, *Pull requests do Gitea*, *Gerrit*, *Subversion* and *Bitbucket Server pull requests*) and *Mercurial* as version control back-ends.

1.11.1 Acessando repositórios

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

Acessando repositórios do Hosted Weblate

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

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

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

Repositórios SSH

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

Aviso: No GitHub, cada chave só pode ser utilizada uma vez, veja *Repositórios do GitHub* e *Acessando repositórios do Hosted Weblate*.

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:

W

Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

⚙️

+

🌐

...

Manage / SSH keys

Webplate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Teams

Appearance

Tools

Automatic suggestions

Billing

Public SSH key

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

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWdxygRSjmbGwJV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhLwpluYlW
Weblate

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-ed25519	+DIY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname Port

Submit

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Chave SSH do Weblate

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

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

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

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

100

Capítulo 1. Documentação de usuário

Verificando chaves SSH do host

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

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

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

Manage / SSH keys

Added host key for github.com with fingerprint p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM (ecdsa-sha2-nistp256), please verify that it is correct.

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

Added host key for github.com with fingerprint +DiY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU (ssh-ed25519), please verify that it is correct.

Weblate status Backups Translation memory Performance report **SSH keys** Alerts Repositories Users Teams

Appearance Tools Automatic suggestions Billing

Public SSH key

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

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWDxygRSjmBGwjV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhLwpl1uYlW
Weblate
```

Download private key

Known host keys

Hostname	Key type	Fingerprint
github.com	ecdsa-sha2-nistp256	p2QAMXNIC1TJYWeIOtrVc98/R1BUFWu3/LiyKgUfQM
github.com	ssh-ed25519	+DiY3wvW6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU
github.com	ssh-rsa	nThbg6kXUpJWGI7E1IGOCspRomTxdCARLviKw6E5SY8

Add host key

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

Hostname Port

Submit

Repositórios do GitHub

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

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

Para implantações menores, use autenticação HTTPS com um token de acesso pessoal e sua conta GitHub, veja [Criando um token de acesso para uso em linha de comando](#).

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

Ver também:

[Acessando repositórios do Hosted Weblate](#)

URLs internas do Weblate

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

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

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

Motivos para usar isso:

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

Repositórios HTTPS

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

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

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

Usando proxy

Se você precisar acessar repositórios VCS por HTTP/HTTPS usando um servidor proxy, configure o VCS para usá-lo. Isto pode ser feito utilizando as variáveis de ambiente `http_proxy`, `https_proxy` e `all_proxy` (como descrito na documentação do [cURL](#)) ou aplicando-a na configuração do VCS, por exemplo:

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

Nota: A configuração do proxy precisa ser feita com o usuário executando Weblate (veja também [Permissões do sistema de arquivos](#)) e com `HOME=$DATA_DIR/home` (veja [DATA_DIR](#)), caso contrário o Git executado pelo Weblate não o utilizará.

Ver também:

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

1.11.2 Git

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

Ver também:

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

Git com push forçado

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

Aviso: Use com cautela, pois isso facilmente leva a commits perdidos em seu repositório upstream.

Personalizando a configuração do Git

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

Auxiliares de remotos do Git

Você também pode usar os [auxiliares de remotos](#) do Git para ter suporte adicionalmente a outros sistemas de controle de versão, mas esteja preparado para depurar problemas que isso pode levar.

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

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

Para clonar o projeto `gnuhello` do Launchpad usando Bazaar:

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

Para o repositório `hello` de `selenic.com` usando Mercurial:

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

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

1.11.3 Pull requests do GitHub

Novo na versão 2.3.

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

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

Você precisa configurar as credenciais da API (*GITHUB_CREDENTIALS*) nas configurações do Weblate para fazer isso funcionar. Uma vez configurado, você verá uma opção *GitHub* ao selecionar *Sistema de controle de versão*.

Ver também:

Fazendo push das alterações do Weblate, GITHUB_CREDENTIALS

1.11.4 Merge requests do GitLab

Novo na versão 3.9.

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

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

Você precisa configurar as credenciais da API (*GITLAB_CREDENTIALS*) nas configurações do Weblate para fazer isso funcionar. Uma vez configurado, você verá uma opção *GitLab* ao selecionar *Sistema de controle de versão*.

Ver também:

Fazendo push das alterações do Weblate, GITLAB_CREDENTIALS

1.11.5 Pull requests do Gitea

Novo na versão 4.12.

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

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

Você precisa configurar as credenciais da API (*GITEA_CREDENTIALS*) nas configurações do Weblate para fazer isso funcionar. Uma vez configurado, você verá uma opção *Gitea* ao selecionar *Sistema de controle de versão*.

Ver também:

Fazendo push das alterações do Weblate, GITEA_CREDENTIALS

1.11.6 Bitbucket Server pull requests

Novo na versão 4.16.

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

Aviso: This does not support Bitbucket Cloud API.

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 [Bitbucket Server pull requests](#) creates pull request.

You need to configure API credentials (`BITBUCKETSERVER_CREDENTIALS`) in the Weblate settings to make this work. Once configured, you will see a [Bitbucket Server](#) option when selecting [Sistema de controle de versão](#).

Ver também:

Fazendo push das alterações do Weblate, `BITBUCKETSERVER_CREDENTIALS`

1.11.7 Merge requests do Pagure

Novo na versão 4.3.2.

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

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

Você precisa configurar as credenciais da API (`PAGURE_CREDENTIALS`) nas configurações do Weblate para fazer isso funcionar. Uma vez configurado, você verá uma opção [Pagure](#) ao selecionar [Sistema de controle de versão](#).

Ver também:

Fazendo push das alterações do Weblate, `PAGURE_CREDENTIALS`

1.11.8 Gerrit

Novo na versão 2.2.

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

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

1.11.9 Mercurial

Novo na versão 2.1.

Mercurial é outro VCS que você pode usar diretamente no Weblate.

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

Ver também:

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

1.11.10 Subversion

Novo na versão 2.8.

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

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

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

Credenciais de Subversion

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

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

Ver também:

`DATA_DIR`

1.11.11 Arquivos locais

1.11.12 Git

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

Novo na versão 3.8.

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

Em segundo plano, o Weblate cria um repositório de Git para você e todas as alterações são rastreadas. Caso você decida mais tarde usar um VCS para armazenar as traduções, você já tem um repositório dentro do Weblate pode basear sua integração.

1.12 API REST do Weblate

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

A API é acessível na URL `/api/` e é baseada em [framework Django REST](#). Você pode utilizá-la diretamente ou com o [Weblate Client](#).

1.12.1 Autenticação e parâmetros genéricos

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

ANY /

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

Parâmetros de consulta

- **format** – Formato de resposta (substitui [Accept](#)). Os valores possíveis dependem da configuração do quadro REST, por padrão `json` e `api` são suportados. Este último fornece uma interface de navegador web para API.
- **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 status 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 usando API
- **web_url** (*string*) – URL para acessar esse recurso usando navegador web

Códigos de status

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

Tokens de autenticação

Alterado na versão 4.10: Tokens com escopo de projeto foram introduzidos na versão 4.10.

Cada usuário possui seu token de acesso pessoal, que pode ser obtido no perfil do usuário. Os tokens de usuário recém-gerados têm o prefixo `wlu_`.

É possível criar tokens com escopo de projeto para acesso à API apenas para determinado projeto. Esses 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 usuários anônimos e 5000 solicitações por hora para usuários autenticados.

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

No contêiner do Docker, isso pode ser configurado usando `WEBLATE_API_RATELIMIT_ANON` e `WEBLATE_API_RATELIMIT_USER`.

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

X-RateLimit-Limit	Limite de limitação de taxa de solicitações para realizar
X-RateLimit-Remaining	Limite restante de solicitações
X-RateLimit-Reset	Número de segundos até que a janela de “ratelimit” seja reiniciada

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

Ver também:

Limitação de taxa, *Limitação de taxa*, `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 Usuários

Novo na versão 4.0.

GET /api/users/

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

Ver também:

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

POST /api/users/

Cria um novo usuário.

Parâmetros

- **username** (*string*) – Nome de usuário
- **full_name** (*string*) – Nome completo do usuário
- **email** (*string*) – E-mail do usuário
- **is_superuser** (*boolean*) – O usuário é superusuário? (opcional)
- **is_active** (*boolean*) – O usuário está ativo? (opcional)
- **is_bot** (*boolean*) – O usuário é bot? (opcional) (usado para tokens no escopo do projeto)

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

Retorna informações sobre os usuários.

Parâmetros

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

Objeto JSON de resposta

- **username** (*string*) – nome de usuário de um usuário
- **full_name** (*string*) – nome completo de um usuário
- **email** (*string*) – e-mail de um usuário
- **is_superuser** (*boolean*) – se o usuário é um superusuário
- **is_active** (*boolean*) – se o usuário está ativo
- **is_bot** (*boolean*) – se o usuário é um bot (usado para tokens no escopo do projeto)

- **date_joined** (*string*) – data de criação do usuário
- **groups** (*array*) – link para grupos associados; veja `GET /api/groups/(int:id)/`

Exemplo de dados JSON:

```
{
  "email": "user@example.com",
  "full_name": "Example User",
  "username": "exampleusername",
  "groups": [
    "http://example.com/api/groups/2/",
    "http://example.com/api/groups/3/"
  ],
  "is_superuser": true,
  "is_active": true,
  "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 usuário.

Parâmetros

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

Objeto JSON de resposta

- **username** (*string*) – nome de usuário de um usuário
- **full_name** (*string*) – nome completo de um usuário
- **email** (*string*) – e-mail de um usuário
- **is_superuser** (*boolean*) – se o usuário é um superusuário
- **is_active** (*boolean*) – se o usuário está ativo
- **is_bot** (*boolean*) – se o usuário é um bot (usado para tokens no escopo do projeto)
- **date_joined** (*string*) – data de criação do usuário

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

Altera os parâmetros do usuário.

Parâmetros

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

Objeto JSON de resposta

- **username** (*string*) – nome de usuário de um usuário
- **full_name** (*string*) – nome completo de um usuário
- **email** (*string*) – e-mail de um usuário
- **is_superuser** (*boolean*) – se o usuário é um superusuário
- **is_active** (*boolean*) – se o usuário está ativo
- **is_bot** (*boolean*) – se o usuário é um bot (usado para tokens no escopo do projeto)
- **date_joined** (*string*) – data de criação do usuário

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

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

Parâmetros

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

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

Associa grupos a um usuário.

Parâmetros

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

Parâmetros de formulário

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

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

Novo na versão 4.13.1.

Remover usuário de um grupo.

Parâmetros

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

Parâmetros de formulário

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

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

Lista as estatísticas de um usuário.

Parâmetros

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

Objeto JSON de resposta

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

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

Lista assinaturas de um usuário.

Parâmetros

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

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

Associa assinaturas com um usuário.

Parâmetros

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

Objeto JSON da solicitação

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

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

Obtém uma assinatura associada a um usuário.

Parâmetros

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

PUT /api/users/ (**str:** *username*) /notifications/
int: *subscription_id*/

Edita uma assinatura associada a um usuário.

Parâmetros

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

Objeto JSON da solicitação

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

PATCH /api/users/ (**str:** *username*) /notifications/
int: *subscription_id*/

Edita uma assinatura associada a um usuário.

Parâmetros

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

Objeto JSON da solicitação

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

DELETE /api/users/ (**str:** *username*) /notifications/
int: *subscription_id*/

Exclui uma assinatura associada a um usuário.

Parâmetros

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

1.12.4 Grupos

Novo na versão 4.0.

GET /api/groups/

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

Ver também:

Os atributos de objeto de grupo estão documentados em [GET /api/groups/\(int:id\)/](#).

POST /api/groups/

Cria um novo grupo.

Parâmetros

- **name** (*string*) – Nome do grupo
- **project_selection** (*int*) – Grupo de seleção de projetos a partir de opções dadas
- **language_selection** (*int*) – Grupo de idiomas selecionados a partir de opções dadas
- **defining_project** (*str*) – link para o projeto de definição, usado para *Gerenciando controle de acesso por projeto*; veja *GET /api/projects/(string:project)/*

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

Retorna informações sobre o grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

- **name** (*string*) – nome de um grupo
- **project_selection** (*int*) – inteiro correspondente ao grupo de projetos
- **language_selection** (*int*) – inteiro correspondente ao grupo de idiomas
- **roles** (*array*) – link para funções associadas; veja *GET /api/roles/(int:id)/*
- **projects** (*array*) – link para projetos associados; veja *GET /api/projects/(string:project)/*
- **components** (*array*) – link para componentes associados; veja *GET /api/components/(string:project)/(string:component)/*
- **componentlists** (*array*) – link para lista de componentes associada; veja *GET /api/component-lists/(str:slug)/*
- **defining_project** (*str*) – link para o projeto de definição, usado para *Gerenciando controle de acesso por projeto*; veja *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/"
  ]
}
```

(continua na próxima página)

(continuação da página anterior)

```
    ]
}
```

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

Altera os parâmetros do grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

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

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

Altera os parâmetros do grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Objeto JSON de resposta

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

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

Exclui o grupo.

Parâmetros

- **id** (*int*) – ID do grupo

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

Associa funções a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

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

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

Associa componentes a um grupo.

Parâmetros

- **id** (*int*) – ID do grupo

Parâmetros de formulário

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

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

Exclui o componente de um grupo.

Parâmetros

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

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

Associa projetos a um grupo.

Parâmetros

- **id** (int) – ID do grupo

Parâmetros de formulário

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

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

int: *project_id*

Exclui o projeto de um grupo.

Parâmetros

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

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

Associa idiomas a um grupo.

Parâmetros

- **id** (int) – ID do grupo

Parâmetros de formulário

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

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

string: *language_code*

Exclui idioma de um grupo.

Parâmetros

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

POST /api/groups/(int: id)/componentlists/

Associa listas de componentes a um grupo.

Parâmetros

- **id** (int) – ID do grupo

Parâmetros de formulário

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

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

int: *component_list_id*

Exclui a lista de componentes de um grupo.

Parâmetros

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

1.12.5 Funções

GET /api/roles/

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

Ver também:

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

POST /api/roles/

Cria uma nova função.

Parâmetros

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

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

Retorna informações sobre uma função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

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

Exemplo de dados JSON:

```
{
  "name": "Access repository",
  "permissions": [
    "vcs.access",
    "vcs.view"
  ],
  "url": "http://example.com/api/roles/1/",
}
```

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

Altera os parâmetros da função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

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

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

Altera os parâmetros da função.

Parâmetros

- **id** (*int*) – ID da função

Objeto JSON de resposta

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

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

Exclui a função.

Parâmetros

- **id** (*int*) – ID da função

1.12.6 Idiomas

GET `/api/languages/`

Retorna uma lista de todos os idiomas.

Ver também:

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

POST `/api/languages/`

Cria um novo idioma.

Parâmetros

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

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

Retorna informações sobre um idioma.

Parâmetros

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

Objeto JSON de resposta

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

Objeto JSON da solicitação

- **population** (*int*) – Número de apresentadores

Exemplo de dados JSON:

```
{
  "code": "en",
  "direction": "ltr",
  "name": "English",
  "population": 159034349015,
  "plural": {
    "id": 75,
    "source": 0,
    "number": 2,
    "formula": "n != 1",
    "type": 1
  },
}
```

(continua na próxima página)

(continuação da página anterior)

```

"aliases": [
    "english",
    "en_en",
    "base",
    "source",
    "eng"
],
"url": "http://example.com/api/languages/en/",
"web_url": "http://example.com/languages/en/",
"statistics_url": "http://example.com/api/languages/en/statistics/"
}

```

PUT /api/languages/ (string: language) /

Altera os parâmetros do idioma.

Parâmetros

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

Objeto JSON da solicitação

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

PATCH /api/languages/ (string: language) /

Altera os parâmetros do idioma.

Parâmetros

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

Objeto JSON da solicitação

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

DELETE /api/languages/ (string: language) /

Exclui o idioma.

Parâmetros

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

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

Retorna estatísticas para um idioma.

Parâmetros

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

Objeto JSON de resposta

- **total** (*int*) – número total de textos
- **total_words** (*int*) – número total de palavras
- **last_change** (*timestamp*) – alterações mais recentes no idioma
- **recent_changes** (*int*) – número total de alterações

- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **translated_words** (*int*) – número de palavras traduzidas
- **translated_words_percent** (*int*) – porcentagem de palavras traduzidas
- **translated_chars** (*int*) – número de caracteres traduzidos
- **translated_chars_percent** (*int*) – porcentagem de caracteres traduzidos
- **total_chars** (*int*) – número total de caracteres
- **fuzzy** (*int*) – número de textos aproximados (marcados para edição)
- **fuzzy_percent** (*int*) – porcentagem de textos aproximados (marcados para edição)
- **failing** (*int*) – número de textos com falha
- **failing** – porcentagem de textos com falha

1.12.7 Projetos

GET /api/projects/

Retorna uma lista de todos os projetos.

Ver também:

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

POST /api/projects/

Novo na versão 3.9.

Cria um novo projeto.

Parâmetros

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

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

Retorna informações sobre um projeto.

Parâmetros

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

Objeto JSON de resposta

- **name** (*string*) – nome do projeto
- **slug** (*string*) – url amigável do projeto
- **web** (*string*) – site do projeto
- **components_list_url** (*string*) – URL para lista de componentes; veja `GET /api/projects/(string:project)/components/`
- **repository_url** (*string*) – URL para status do repositório; veja `GET /api/projects/(string:project)/repository/`
- **changes_list_url** (*string*) – URL para lista de alterações; veja `GET /api/projects/(string:project)/changes/`
- **translation_review** (*boolean*) – *Habilitar revisões*

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

Exemplo de dados JSON:

```
{
  "name": "Hello",
  "slug": "hello",
  "url": "http://example.com/api/projects/hello/",
  "web": "https://weblate.org/",
  "web_url": "http://example.com/projects/hello/"
}
```

PATCH /api/projects/ (*string: project*) /

Novo na versão 4.3.

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

Parâmetros

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

PUT /api/projects/ (*string: project*) /

Novo na versão 4.3.

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

Parâmetros

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

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

Novo na versão 3.9.

Exclui um projeto.

Parâmetros

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

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

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

Parâmetros

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

Objeto JSON de resposta

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

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

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

Parâmetros

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

Objeto JSON de resposta

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

Exemplo de dados JSON:

```
{
  "needs_commit": true,
  "needs_merge": false,
  "needs_push": true
}
```

POST /api/projects/(string: *project*)/repository/

Realiza a operação dada no repositório VCS.

Parâmetros

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

Objeto JSON da solicitação

- **operation** (*string*) – Operação para realizar: uma de push, pull, commit, reset, cleanup, file-sync

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{"operation": "pull"}
```

Exemplo de resposta JSON:

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

{"result": true}
```

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

Retorna uma lista de componentes de tradução no projeto dado.

Parâmetros

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos componentes; veja `GET /api/components/(string:project)/(string:component)/`

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

Novo na versão 3.9.

Alterado na versão 4.3: Os parâmetros `zipfile` e `docfile` agora são aceitos para componentes sem VCS, veja *Arquivos locais*.

Alterado na versão 4.6: Os repositórios clonados agora são automaticamente compartilhados dentro de um projeto usando *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 amigável do projeto

Parâmetros de formulário

- **file zipfile** – Arquivo ZIP para carregar no Weblate para inicialização de traduções
- **file docfile** – Documento para traduzir
- **boolean disable_autoshare** – Desabilita 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 arquivos usando os parâmetros `zipfile` e `docfile`. Os dados devem ser enviados como *multipart/form-data*.

Exemplo de solicitação de formulário com CURL:

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

Exemplo de solicitação JSON com CURL:

```
curl \
  --data-binary '{
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "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 novo 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 novo 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",
  "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
```

(continua na próxima página)

(continuação da página anterior)

```

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 em um projeto.

Novo na versão 3.8.

Parâmetros

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de estatísticas de tradução
- **language** (*string*) – nome de idioma
- **code** (*string*) – código de idioma
- **total** (*int*) – número total de textos
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **total_words** (*int*) – número total de palavras
- **translated_words** (*int*) – número de palavras traduzidas

- **words_percent** (*float*) – porcentagem de palavras traduzidas

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

Retorna estatísticas para um projeto.

Novo na versão 3.8.

Parâmetros

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

Objeto JSON de resposta

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

1.12.8 Componentes

Dica: Usa `POST /api/projects/(string:project)/repository/` para criar novos componentes.

GET `/api/components/`

Retorna uma lista de componentes de tradução.

Ver também:

Os atributos do objeto componente estão documentados em `GET /api/components/(string:project)/(string:component)/`.

GET `/api/components/(string: project) /
string: component/`

Retorna informações sobre o componente de tradução.

Parâmetros

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

Objeto JSON de resposta

- **project** (*object*) – o projeto de tradução; veja `GET /api/projects/(string:project)/`
- **name** (*string*) – *Nome do componente*
- **slug** (*string*) – *URL amigável do componente*
- **vcs** (*string*) – *Sistema de controle de versão*
- **repo** (*string*) – *Repositório do código-fonte*
- **git_export** (*string*) – *URL do repositório exportado*
- **branch** (*string*) – *Ramo do repositório*
- **push_branch** (*string*) – *Ramo do push*
- **filemask** (*string*) – *Máscara de arquivo*
- **template** (*string*) – *Arquivo de idioma da base monolíngue*

- **edit_template** (*string*) – Editar o arquivo base
- **intermediate** (*string*) – Arquivo de idioma intermediário
- **new_base** (*string*) – Modelo para novas traduções
- **file_format** (*string*) – Formato de arquivo
- **license** (*string*) – Licença da tradução
- **agreement** (*string*) – Acordo de colaborador
- **new_lang** (*string*) – Adicionando nova tradução
- **language_code_style** (*string*) – Estilo de código de idioma
- **source_language** (*object*) – objeto do idioma fonte; veja `GET /api/languages/(string:language)/`
- **push** (*string*) – URL de push do repositório
- **check_flags** (*string*) – Marcadores de tradução
- **priority** (*string*) – Prioridade
- **enforced_checks** (*string*) – Verificações forçadas
- **restricted** (*string*) – Acesso restrito
- **repoweb** (*string*) – Navegador do repositório
- **report_source_bugs** (*string*) – Endereço do relatório de erros do texto fonte
- **merge_style** (*string*) – Estilo de mesclagem
- **commit_message** (*string*) – Mensagens de commit, add, merge, extensão e merge request
- **add_message** (*string*) – Mensagens de commit, add, merge, extensão e merge request
- **delete_message** (*string*) – Mensagens de commit, add, merge, extensão e merge request
- **merge_message** (*string*) – Mensagens de commit, add, merge, extensão e merge request
- **addon_message** (*string*) – Mensagens de commit, add, merge, extensão e merge request
- **pull_message** (*string*) – Mensagens de commit, add, merge, extensão e merge request
- **allow_translation_propagation** (*string*) – Permitir propagação de tradução
- **enable_suggestions** (*string*) – Habilitar sugestões
- **suggestion_voting** (*string*) – Votação de sugestões
- **suggestion_autoaccept** (*string*) – Aceitar sugestões automaticamente
- **push_on_commit** (*string*) – Push ao fazer commit
- **commit_pending_age** (*string*) – Idade das alterações para fazer commit
- **auto_lock_error** (*string*) – Bloquear em erro
- **language_regex** (*string*) – Filtro de idioma
- **variant_regex** (*string*) – Expressão regular de variantes
- **repository_url** (*string*) – URL para status do repositório; veja `GET /api/components/(string:project)/(string:component)/repository/`

- **translations_url** (*string*) – URL para lista de tradução; veja `GET /api/components/(string:project)/(string:component)/translations/`
- **lock_url** (*string*) – URL para status de bloqueio; veja `GET /api/components/(string:project)/(string:component)/lock/`
- **changes_list_url** (*string*) – URL para lista de alterações; veja `GET /api/components/(string:project)/(string:component)/changes/`
- **task_url** (*string*) – URL para uma tarefa em segundo plano (se houver); veja `GET /api/tasks/(str:uuid)/`

Exemplo de dados JSON:

```
{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "Weblate",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "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 amigável do projeto
- **component** (*string*) – URL amigável do componente

- **source_language** (*string*) – Código de idioma fonte do projeto (opcional)

Objeto JSON da solicitação

- **name** (*string*) – nome do componente
- **slug** (*string*) – URL amigável do componente
- **repo** (*string*) – URL do repositório VCS

Exemplo de CURL:

```
curl \
  --data-binary '{"name": "new name"}' \
  -H "Content-Type: application/json" \
  -H "Authorization: Token TOKEN" \
  PATCH http://example.com/api/projects/hello/components/
```

Exemplo de solicitação JSON:

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

{
  "name": "new name"
}
```

Exemplo de resposta JSON:

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

{
  "branch": "main",
  "file_format": "po",
  "filemask": "po/*.po",
  "git_export": "",
  "license": "",
  "license_url": "",
  "name": "new name",
  "slug": "weblate",
  "project": {
    "name": "Hello",
    "slug": "hello",
    "source_language": {
      "code": "en",
      "direction": "ltr",
      "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/",
```

(continua na próxima página)

(continuação da página anterior)

```
    "web_url": "http://example.com/projects/hello/"
  },
  "repo": "file:///home/nijel/work/weblate-hello",
  "template": "",
  "new_base": "",
  "url": "http://example.com/api/components/hello/weblate/",
  "vcs": "git",
  "web_url": "http://example.com/projects/hello/weblate/"
}
```

PUT `/api/components/(string: project) /`
`string: component/`

Edita um componente por uma solicitação PUT.

Parâmetros

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

Objeto JSON da solicitação

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

DELETE `/api/components/(string: project) /`
`string: component/`

Novo na versão 3.9.

Exclui um componente.

Parâmetros

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

GET `/api/components/(string: project) /`
`string: component/changes/`

Retorna uma lista de alterações de componentes. Este é essencialmente um componente com escopo `GET /api/changes/` aceitando os mesmos parâmetros.

Parâmetros

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

Objeto JSON de resposta

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

GET `/api/components/(string: project) /`
`string: component/file/`

Novo na versão 4.9.

Baixa todas as traduções disponíveis associadas ao componente como um arquivo usando o formato solicitado.

Parâmetros

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

Parâmetros de consulta

- **format** (*string*) – O formato de arquivo 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 tela do componente.

Parâmetros

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

Objeto JSON de resposta

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

GET /api/components/ (**string:** *project*) /
string: *component/lock/*

Retorna o status do bloqueio do componente.

Parâmetros

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

Objeto JSON de resposta

- **locked** (*boolean*) – se o componente está bloqueado para atualizações

Exemplo de dados JSON:

```
{
  "locked": false
}
```

POST /api/components/ (**string:** *project*) /
string: *component/lock/*

Define o status de bloqueio do componente.

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

Parâmetros

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

Objeto JSON da solicitação

- **lock** – Booleano se deve ser bloqueado ou não.

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{"lock": true}
```

Exemplo de resposta JSON:

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

{"locked": true}
```

GET `/api/components/(string: project) /`
`string: component/repository/`

Retorna informações sobre o status do repositório VCS.

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

Parâmetros

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

Objeto JSON de resposta

- **needs_commit** (*boolean*) – se há alguma alteração pendente de commit
- **needs_merge** (*boolean*) – se há alguma alteração do upstream para mesclar
- **needs_push** (*boolean*) – se há alguma alteração local para fazer push
- **remote_commit** (*string*) – Informações do commit remoto
- **status** (*string*) – Status do repositório VCS como relatado pelo VCS
- **merge_failure** – Texto descrevendo uma falha de mesclagem ou nulo se houver nenhuma

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

Executa a operação fornecida em um repositório VCS.

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

Parâmetros

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

Objeto JSON da solicitação

- **operation** (*string*) – Operação para executar: um de push, pull, commit, reset, cleanup

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

Exemplo de CURL:

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

Exemplo de solicitação JSON:

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

{"operation": "pull"}
```

Exemplo de resposta JSON:

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

{"result": true}
```

GET /api/components/(string: *project*) /
string: *component/monolingual_base/*

Faz o download do arquivo base para traduções monolíngues.

Parâmetros

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

GET /api/components/(string: *project*) /
string: *component/new_template/*

Faz o download do arquivo de modelo para novas traduções.

Parâmetros

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

GET /api/components/(string: *project*) /
string: *component/translations/*

Retorna uma lista de objetos de tradução no componente fornecido.

Parâmetros

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

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de tradução; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/`

**POST /api/components/(string: project) /
string: component/translations/**

Cria uma nova tradução no componente fornecido.

Parâmetros

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

Objeto JSON da solicitação

- **language_code** (*string*) – código de idioma de tradução; veja `GET /api/languages/(string:language)/`

Objeto JSON de resposta

- **result** (*object*) – novo objeto de tradução criado

Exemplo de CURL:

```
curl \
  -d language_code=cs \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/projects/hello/components/
```

Exemplo de solicitação JSON:

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

{"language_code": "cs"}
```

Exemplo de resposta JSON:

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

{
  "failing_checks": 0,
  "failing_checks_percent": 0,
  "failing_checks_words": 0,
  "filename": "po/cs.po",
  "fuzzy": 0,
  "fuzzy_percent": 0.0,
  "fuzzy_words": 0,
  "have_comment": 0,
  "have_suggestion": 0,
  "is_template": false,
```

(continua na próxima página)

(continuação da página anterior)

```

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

Objeto JSON de resposta

- **results** (*array*) – vetor de objetos de estatísticas de tradução; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/`

GET `/api/components/(string: project) /`
string: `component/links/`

Retorna projetos vinculados a um componente.

Novo na versão 4.5.

Parâmetros

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

Objeto JSON de resposta

- **projects** (*array*) – projetos associados; veja `GET /api/projects/(string:project)/`

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

Associa projeto a um componente.

Novo na versão 4.5.

Parâmetros

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

Parâmetros de formulário

- **string project_slug** – URL amigável do projeto

DELETE /api/components/ (**string:** *project*) /
string: *component/links/string:* *project_slug/*

Remove associação de um projeto com um componente.

Novo na versão 4.5.

Parâmetros

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

1.12.9 Traduções

GET /api/translations/

Retorna uma lista de traduções.

Ver também:

Atributos de objeto de tradução estão documentados em [GET /api/translations/ \(*string:project*\)/ \(*string:component*\)/ \(*string:language*\)/](#).

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

Retorna informações sobre uma tradução.

Parâmetros

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

Objeto JSON de resposta

- **component** (*object*) – objeto do componente; veja [GET /api/components/ \(*string:project*\)/ \(*string:component*\)/](#)
- **failing_checks** (*int*) – número de textos contendo verificações com falha
- **failing_checks_percent** (*float*) – porcentagem de textos contendo verificações com falha
- **failing_checks_words** (*int*) – número de palavras contendo verificações com falha
- **filename** (*string*) – Nome de arquivo da tradução
- **fuzzy** (*int*) – número de textos aproximados (marcados para edição)
- **fuzzy_percent** (*float*) – porcentagem de textos aproximados (marcados para edição)
- **fuzzy_words** (*int*) – número de palavras em textos aproximados (marcados para edição)
- **have_comment** (*int*) – número de textos com comentário
- **have_suggestion** (*int*) – número de textos com sugestão

- **is_template** (*boolean*) – se a tradução tem uma base monolíngue
- **language** (*object*) – objeto do idioma fonte; veja `GET /api/languages/(string:language)/`
- **language_code** (*string*) – código de idioma usado no repositório; isso pode ser diferente do código de idioma no objeto de linguagem
- **last_author** (*string*) – nome do último autor
- **last_change** (*timestamp*) – data e hora da última alteração
- **revision** (*string*) – hash de revisão para o arquivo
- **share_url** (*string*) – URL para compartilhar levando à página de engajamento
- **total** (*int*) – número total de textos
- **total_words** (*int*) – número total de palavras
- **translate_url** (*string*) – URL para tradução
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **translated_words** (*int*) – número de palavras traduzidas
- **repository_url** (*string*) – URL para status do repositório; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/repository/`
- **file_url** (*string*) – URL para objeto arquivo; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/file/`
- **changes_list_url** (*string*) – URL para lista de alterações; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/changes/`
- **units_list_url** (*string*) – URL para lista de textos; veja `GET /api/translations/(string:project)/(string:component)/(string:language)/units/`

Exemplo de dados JSON:

```
{
  "component": {
    "branch": "main",
    "file_format": "po",
    "filemask": "po/*.po",
    "git_export": "",
    "license": "",
    "license_url": "",
    "name": "Weblate",
    "new_base": "",
    "project": {
      "name": "Hello",
      "slug": "hello",
      "source_language": {
        "code": "en",
        "direction": "ltr",
        "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/"
    }
  }
}
```

(continua na próxima página)

```

        "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.

Exclui uma tradução.

Parâmetros

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

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

Retorna uma lista de alterações de tradução. Este é essencialmente um `GET /api/changes/` no escopo de tradução aceitando os mesmos parâmetros.

Parâmetros

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

Objeto JSON de resposta

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

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

Retorna uma lista de unidades de tradução.

Parâmetros

- **project** (*string*) – URL amigável do projeto
- **component** (*string*) – URL amigável do componente
- **language** (*string*) – Código de idioma de tradução
- **q** (*string*) – Texto de consulta de pesquisa; veja [Pesquisando](#) (opcional)

Objeto JSON de resposta

- **results** (*array*) – texto de objetos de componente; veja [GET /api/units/\(int:id\)/](#)

POST /api/translations/(string: project) /
string: component/string: language/units/

Adiciona nova unidade.

Parâmetros

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

Objeto JSON da solicitação

- **key** (*string*) – Nome da unidade de tradução (usada como chave ou contexto)
- **value** (*array*) – Textos fonte (use um único texto se não estiver criando plural)
- **state** (*int*) – String state; see [GET /api/units/\(int:id\)/](#)

Objeto JSON de resposta

- **unit** (*object*) – unidade recém-criada; veja [GET /api/units/\(int:id\)/](#)

Ver também:

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

POST /api/translations/(string: project) /
string: component/string: language/autotranslate/

Aciona a tradução automática.

Parâmetros

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

Objeto JSON da solicitação

- **mode** (*string*) – Modo de tradução automática

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

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

Baixa o arquivo de tradução atual como está armazenado no VCS (sem o parâmetro `format`) ou convertido para outro formato (veja [Baixando traduções](#)).

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

Parâmetros de consulta

- **format** – File format to use; if not specified no format conversion happens; supported file formats: `po`, `mo`, `xliff`, `xliff11`, `tbx`, `tmx`, `csv`, `xlsx`, `json`, `aresource`, `strings`
- **q** (*string*) – Filtre as strings baixadas, consulte `search`, aplicável somente quando a conversão estiver em vigor (`format` estiver especificado).

Parâmetros

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

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

Envia novo arquivo com traduções.

Parâmetros

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

Parâmetros de formulário

- **string conflicts** – Como lidar com conflitos (`ignore`, `replace-translated` ou `replace-approved`)
- **file file** – Arquivo enviado
- **string email** – E-mail do autor
- **string author** – Nome do autor
- **string method** – Método de envio (`translate`, `approve`, `suggest`, `fuzzy`, `replace`, `source`, `add`), veja [Métodos de importação](#)
- **string fuzzy** – Processamento de textos aproximados (marcados para edição) (`vazio`, `process`, `approve`)

Exemplo de CURL:


```
curl -X POST \
  -F file=@strings.xml \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/translations/hello/android/cs/file/
```

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

Retorna informações sobre o status do repositório VCS.

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

Parâmetros

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

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

Realiza a operação dada no repositório VCS.

Vea `POST /api/projects/(string:project) / repository/` para a documentação.

Parâmetros

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

Objeto JSON da solicitação

- **operation** (*string*) – Operação para executar: um de push, pull, commit, re-set, cleanup

Objeto JSON de resposta

- **result** (*boolean*) – resultado da operação

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

Retorna as estatísticas de tradução detalhadas.

Novo na versão 2.7.

Parâmetros

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

Objeto JSON de resposta

- **code** (*string*) – código de idioma
- **failing** (*int*) – número de verificações com falha
- **failing_percent** (*float*) – porcentagem de verificações com falha
- **fuzzy** (*int*) – número de textos aproximados (marcados para edição)
- **fuzzy_percent** (*float*) – porcentagem de textos aproximados (marcados para edição)
- **total_words** (*int*) – número total de palavras

- **translated_words** (*int*) – número de palavras traduzidas
- **last_author** (*string*) – nome do último autor
- **last_change** (*timestamp*) – data da última alteração
- **name** (*string*) – nome de idioma
- **total** (*int*) – número total de textos
- **translated** (*int*) – número de textos traduzidos
- **translated_percent** (*float*) – porcentagem de textos traduzidos
- **url** (*string*) – URL para acesso à tradução (URL de engajamento)
- **url_translate** (*string*) – URL para acesso à tradução (URL real da tradução)

1.12.10 Memória

Novo na versão 4.14.

GET /api/memory/

Retorna uma lista de resultados de memória.

DELETE /api/memory/ (*int: memory_object_id*) /

Exclui um objeto de memória

Parâmetros

- **memory_object_id** – ID de Objeto de Memória

1.12.11 Unidades

Uma *unidade* é uma única parte de uma tradução que forma o par de um texto fonte com um texto traduzido correspondente e também contém alguns metadados relacionados. O termo é derivado do [Translate Toolkit](#) e do XLIFF.

Novo na versão 2.10.

GET /api/units/

Retorna uma lista de unidades de tradução.

Parâmetros

- **q** (*string*) – Texto de consulta de pesquisa; veja [Pesquisando](#) (opcional)

Ver também:

Os atributos do objeto da unidade estão documentados em [GET /api/units/\(int:id\)/](#).

GET /api/units/ (*int: id*) /

Alterado na versão 4.3: O `target` e `source` agora são vetores para lidar apropriadamente com textos plurais.

Retorna informações sobre a unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON de resposta

- **translation** (*string*) – URL de um objeto de tradução relacionado
- **source** (*array*) – texto fonte
- **previous_source** (*string*) – texto fonte anterior usado para correspondência aproximada

- **target** (*array*) – texto alvo
- **id_hash** (*string*) – identificador único da unidade
- **content_hash** (*string*) – identificador único do texto fonte
- **location** (*string*) – local da unidade no código-fonte
- **context** (*string*) – contexto da unidade de tradução
- **note** (*string*) – nota da unidade de tradução
- **flags** (*string*) – marcadores da unidade de tradução
- **labels** (*array*) – rótulos de unidade de tradução, disponíveis em unidades fonte
- **state** (*int*) – estado da unidade, 0 - não traduzida, 10 - necessita tradução, 20 - traduzida, 30 - aprovada, 100 - somente leitura
- **fuzzy** (*boolean*) – se a unidade está marcada com “fuzzy” ou para revisão
- **translated** (*boolean*) – se a unidade está traduzida
- **approved** (*boolean*) – se a tradução está aprovada
- **position** (*int*) – posição da unidade no arquivo de tradução
- **has_suggestion** (*boolean*) – se a unidade tem sugestões
- **has_comment** (*boolean*) – se a unidade tem comentários
- **has_failing_check** (*boolean*) – se a unidade tem verificações com falha
- **num_words** (*int*) – número de palavras fonte
- **priority** (*int*) – prioridade de tradução. 100 é o padrão
- **id** (*int*) – identificador da unidade
- **explanation** (*string*) – Expansão de texto, disponível em unidades fonte, veja [Informações adicionais sobre textos fonte](#)
- **extra_flags** (*string*) – Marcadores de textos adicionais, disponíveis nas unidades fonte, veja [Personalizando comportamento usando marcadores](#)
- **web_url** (*string*) – URL onde a unidade pode ser editada
- **source_unit** (*string*) – Link da unidade fonte; veja `GET /api/units/(int:id)/`
- **pending** (*boolean*) – se a unidade está pendente para escrita
- **timestamp** (*timestamp*) – idade do texto

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

Novo na versão 4.3.

Executa atualização parcial na unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON da solicitação

- **state** (*int*) – estado da unidade, 0 - não traduzida, 10 - necessita edição, 20 - traduzida, 30 - aprovada (fluxo de trabalho de exigência de revisão habilitado, veja [Revisores dedicados](#))
- **target** (*array*) – texto alvo
- **explanation** (*string*) – Expansão de texto, disponível em unidades fonte, veja [Informações adicionais sobre textos fonte](#)

- **extra_flags** (*string*) – Marcadores de textos adicionais, disponíveis nas unidades fonte, veja [Personalizando comportamento usando marcadores](#)

Objeto JSON de resposta

- **labels** (*array*) – rótulos, disponíveis nas unidades fonte

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

Novo na versão 4.3.

Executa atualização completa na unidade de tradução.

Parâmetros

- **id** (*int*) – ID da unidade

Objeto JSON da solicitação

- **state** (*int*) – estado da unidade, 0 - não traduzida, 10 - necessita edição, 20 - traduzida, 30 - aprovada (fluxo de trabalho de exigência de revisão habilitado, veja [Revisores dedicados](#))
- **target** (*array*) – texto alvo
- **explanation** (*string*) – Expansão de texto, disponível em unidades fonte, veja [Informações adicionais sobre textos fonte](#)
- **extra_flags** (*string*) – Marcadores de textos adicionais, disponíveis nas unidades fonte, veja [Personalizando comportamento usando marcadores](#)

Objeto JSON de resposta

- **labels** (*array*) – rótulos, disponíveis nas unidades fonte

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

Novo na versão 4.3.

Exclui uma 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.

Ver também:

Os atributos do objeto de alteração estão documentados em [GET /api/changes/\(int:id\)/](#).

Parâmetros de consulta

- **user** (*string*) – Nome de usuário do usuário para filtrar
- **action** (*int*) – Ação para filtrar, pode ser usada várias vezes
- **timestamp_after** (*timestamp*) – Data e hora no formato ISO 8601 para listar as alterações após
- **timestamp_before** (*timestamp*) – Data e hora no formato ISO 8601 para listar as alterações antes

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

Retorna informações sobre alteração de tradução.

Parâmetros

- **id** (*int*) – ID da alteração

Objeto JSON de resposta

- **unit** (*string*) – URL de um objeto de unidade relacionado
- **translation** (*string*) – URL de um objeto de tradução relacionado
- **component** (*string*) – URL de um objeto de componente relacionado
- **user** (*string*) – URL de um objeto de usuário relacionado
- **author** (*string*) – URL de um objeto de autor relacionado
- **timestamp** (*timestamp*) – data e hora do evento
- **action** (*int*) – identificação numérica da ação
- **action_name** (*string*) – descrição de texto da ação
- **target** (*string*) – texto ou detalhe alterado de evento
- **id** (*int*) – identificador da alteração

1.12.13 Capturas de tela

Novo na versão 2.14.

GET `/api/screenshots/`

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

Ver também:

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

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

Retorna informações sobre dados de captura de tela.

Parâmetros

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

Objeto JSON de resposta

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

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

Baixa a imagem de captura de tela.

Parâmetros

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

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

Substitui a imagem de captura de tela.

Parâmetros

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

Parâmetros de formulário

- **file image** – Arquivo enviado

Exemplo de CURL:

```
curl -X POST \
  -F image=@image.png \
  -H "Authorization: Token TOKEN" \
  http://example.com/api/screenshots/1/file/
```

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

Associa texto fonte com uma captura de tela.

Parâmetros

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

Parâmetros de formulário

- **string unit_id** – ID da unidade

Objeto JSON de resposta

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

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

int: *unit_id*

Remove a associação de texto fonte com captura de tela.

Parâmetros

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

POST /api/screenshots/

Cria uma nova captura de tela.

Parâmetros de formulário

- **file image** – Arquivo enviado
- **string name** – Nome da captura de tela
- **string project_slug** – URL amigável do projeto
- **string component_slug** – URL amigável do componente
- **string language_code** – Código do idioma

Objeto JSON de resposta

- **name** (*string*) – nome de uma captura de tela
- **component** (*string*) – URL de um objeto de componente relacionado

- **file_url** (*string*) – URL para baixar um arquivo; veja `GET /api/screenshots/(int:id)/file/`
- **units** (*array*) – link para associar informações sobre texto fonte; veja `GET /api/units/(int:id)/`

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

Edita informações parciais sobre captura de tela.

Parâmetros

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

Objeto JSON de resposta

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

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

Edita informações completas sobre captura de tela.

Parâmetros

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

Objeto JSON de resposta

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

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

Exclui captura de tela.

Parâmetros

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

1.12.14 Extensões

Novo na versão 4.4.1.

GET /api/addons/

Retorna uma lista de extensões.

Ver também:

Os atributos do objeto de extensão estão documentados em `GET /api/addons/(int:id)/`.

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

Retorna informações sobre dados da extensão.

Parâmetros

- **id** (*int*) – ID da extensão

Objeto JSON de resposta

- **name** (*string*) – nome de uma extensão
- **component** (*string*) – URL de um objeto de componente relacionado
- **configuration** (*object*) – Configuração opcional da extensão

Ver 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 dos componentes

Novo na versão 4.0.

GET /api/component-lists/

Retorna uma lista de listas de componentes.

Ver também:

Os atributos do objeto da lista de componentes estão documentados em *GET* /api/component-lists/ (*str*:*slug*)/.

GET `/api/component-lists/ (str: slug) /`

Retorna informações sobre a lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON de resposta

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar em um painel
- **components** (*array*) – link para componentes associados; veja `GET /api/components/(string:project)/(string:component)/`
- **auto_assign** (*array*) – regras de atribuição automáticas

PUT `/api/component-lists/ (str: slug) /`

Altera os parâmetros da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON da solicitação

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar em um painel

PATCH `/api/component-lists/ (str: slug) /`

Altera os parâmetros da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Objeto JSON da solicitação

- **name** (*string*) – nome de uma lista de componentes
- **slug** (*string*) – URL amigável de uma lista de componentes
- **show_dashboard** (*boolean*) – se deve mostrar em um painel

DELETE `/api/component-lists/ (str: slug) /`

Exclui a lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

POST `/api/component-lists/ (str: slug) /components/`

Associa um componente a uma lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes

Parâmetros de formulário

- **string component_id** – ID do componente

DELETE `/api/component-lists/ (str: slug) /components/`

str: *component_slug*

Desassocia um componente da lista de componentes.

Parâmetros

- **slug** (*string*) – URL amigável da lista de componentes
- **component_slug** (*string*) – URL amigável do componente

1.12.16 Glossário

Alterado na versão 4.5: Glossários agora são armazenados como componentes, traduções e textos comuns. Use a respectiva 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 porcentagem
- **result** (*object*) – Resultado da tarefa ou detalhes do progresso
- **log** (*string*) – Registros da tarefa

1.12.18 Métricas

GET /api/metrics/

Retorna as métricas do servidor.

Objeto JSON de resposta

- **units** (*int*) – Número de unidades
- **units_translated** (*int*) – Número de unidades traduzidas
- **users** (*int*) – número de usuários
- **changes** (*int*) – Quantidade de alterações
- **projects** (*int*) – Número de projetos
- **components** (*int*) – Número de componentes
- **translations** (*int*) – Número de traduções
- **languages** (*int*) – Quantidade de idiomas usados
- **checks** (*int*) – Número de verificações de qualidade acionadas
- **configuration_errors** (*int*) – Número de erros de configuração
- **suggestions** (*int*) – Número de sugestões pendentes
- **celery_queues** (*object*) – Comprimento das filas de Celery, veja [Tarefas de fundo usando Celery](#)
- **name** (*string*) – Nome do servidor configurado

1.12.19 Ganchos de notificação

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

Você pode usar terminais de repositório para projetos, componentes e traduções para atualizar repositórios individuais; veja `POST /api/projects/(string:project)/repository/` para a documentação.

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

Obsoleto desde a versão 2.6: Use `POST /api/components/(string:project)/(string:component)/repository/` em vez disso, que funciona corretamente com autenticação para projetos limitados por ACL.

Aciona a atualização de um componente (baixando do VCS e procurando por alterações de tradução).

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

Obsoleto desde a versão 2.6: Use `POST /api/projects/(string:project)/repository/`, que funciona corretamente com autenticação para projetos limitados por ACL.

Aciona a atualização de todos os componentes em um projeto (baixando do VCS e procurando por alterações de tradução).

POST /hooks/github/

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

Nota: O GitHub inclui suporte direto para notificar o Weblate: habilite o gancho do serviço Weblate nas configurações do repositório e defina a URL para a URL de sua instalação do Weblate.

Ver também:

Recebendo automaticamente alterações do GitHub

Para instruções sobre como configurar integração com GitHub

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

Informações genéricas sobre Webhooks do GitHub

ENABLE_HOOKS

Para habilitar ganchos para todo o Weblate

POST /hooks/gitlab/

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

Ver também:

Recebendo automaticamente alterações do GitLab

Para instrução sobre configurar integração com GitLab

<https://docs.gitlab.com/ee/user/project/integrations/webhooks.html>

Informações genéricas sobre Webhooks do GitLab

ENABLE_HOOKS

Para habilitar ganchos para todo o Weblate

POST /hooks/bitbucket/

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

Ver também:

Recebendo automaticamente alterações do Bitbucket

Para instruções sobre como configurar integração com Bitbucket

<https://support.atlassian.com/bitbucket-cloud/docs/manage-webhooks/>

Informações genéricas sobre Webhooks do Bitbucket

ENABLE_HOOKS

Para habilitar ganchos para todo o Weblate

POST /hooks/pagure/

Novo na versão 3.3.

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

Ver também:

Recebendo automaticamente alterações do Pagure

Para instruções sobre como configurar integração com Pagure

https://docs.pagure.org/pagure/usage/using_webhooks.html

Informações genéricas sobre Webhooks do Pagure

ENABLE_HOOKS

Para habilitar ganchos para todo o Weblate

POST /hooks/azure/

Novo na versão 3.8.

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

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.

Ver também:

Recebendo automaticamente alterações do Azure Repos

Para instruções sobre como configurar integração com Azure

https:

[//learn.microsoft.com/pt-br/azure/devops/service-hooks/services/webhooks?view=azure-devops](https://learn.microsoft.com/pt-br/azure/devops/service-hooks/services/webhooks?view=azure-devops)

Informações genéricas sobre Web Hooks do Azure DevOps

ENABLE_HOOKS

Para habilitar ganchos para todo o Weblate

POST /hooks/gitea/

Novo na versão 3.9.

Gancho especial para lidar com notificações de Webhook do Gitea e atualizar automaticamente os componentes correspondentes.

Ver também:

Recebendo automaticamente alterações do Gitea Repos

Para instruções sobre como configurar integração com Gitea

<https://docs.gitea.io/pt-br/webhooks/>

Informações genéricas sobre Webhooks do Gitea

ENABLE_HOOKS

Para habilitar ganchos para todo o Weblate

POST /hooks/gitee/

Novo na versão 3.9.

Gancho especial para lidar com notificações de Webhook do Gitee e atualizar automaticamente os componentes correspondentes.

Ver também:

Recebendo automaticamente alterações de Gitee Repos

Para instruções sobre como configurar integração com Gitee

<https://gitee.com/help/categories/40>

Informações genéricas sobre Webhooks do Gitee

ENABLE_HOOKS

Para habilitar ganchos para todo o Weblate

1.12.20 Exportações

O Weblate fornece várias exportações para permitir que você processe ainda mais os dados.

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

Parâmetros de consulta

- **format** (*string*) – Formato de saída: json ou csv

Obsoleto desde a versão 2.6: Use `GET /api/components/(string:project)/(string:component)/statistics/` e `GET /api/translations/(string:project)/(string:component)/(string:language)/statistics/` em vez disso; ele também permite acesso a projetos controlados por ACL.

Recupera estatísticas para determinado componente em determinado formato.

Exemplo de solicitação:

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

Exemplo de resposta:

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

[
  {
    "code": "cs",
    "failing": 0,
    "failing_percent": 0.0,
    "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/",
```

(continua na próxima página)

(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.

Ver também:

RSS na Wikipédia

1.13 Weblate Client

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

1.13.1 Instalação

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

```
pip install wlc
```

1.13.2 Uso com Docker

O Weblate Client também está disponível como uma imagem Docker.

A imagem está publicada no Docker Hub: <https://hub.docker.com/r/weblate/wlc>

Instalação:

```
docker pull weblate/wlc
```

O contêiner Docker usa as configurações padrão do Weblate e se conecta à API implementada no localhost. A URL da API e API_KEY podem ser configuradas através dos argumentos aceitos pelo Weblate.

O comando para iniciar o contêiner usa a seguinte sintaxe:

```
docker run --rm weblate/wlc [WLC_ARGS]
```

Exemplo:

```
docker run --rm weblate/wlc --url https://hosted.weblate.org/api/ list-projects
```

Você pode querer passar seu *Arquivos de configuração* para o contêiner Docker, a abordagem mais fácil é adicionar seu diretório atual como o volume /home/weblate:

```
docker run --volume $PWD:/home/weblate --rm weblate/wlc show
```

1.13.3 Primeiros Passos

A configuração do wlc é armazenada em ~/.config/weblate (consulte *Arquivos de configuração* para outros locais). Crie-o para corresponder ao seu ambiente:

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

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

Em seguida, você pode invocar comandos no servidor padrão:

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

Ver também:

Arquivos de configuração

1.13.4 Sinopse

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

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

1.13.5 Descrição

Weblate Client é uma biblioteca Python e utilitário de linha de comando para gerenciar o Weblate remotamente usando a *API REST do Weblate*. O utilitário de linha de comando pode ser invocado como **wlc** e está embutido em *wlc*.

Argumentos

O programa aceita os seguintes argumentos que definem o formato de saída ou qual instância do Weblate usar. Estes devem ser inseridos antes de qualquer comando.

--format {csv,json,text,html}

Especifica o formato de saída.

--url URL

Especifica a URL da API. Substitui qualquer valor encontrado no arquivo de configuração, consulte *Arquivos de configuração*. A URL deve terminar com /api/, por exemplo, `https://hosted.weblate.org/api/`.

--key KEY

Especifica a chave do usuário de API a ser usada. Substitui qualquer valor encontrado no arquivo de configuração, consulte *Arquivos de configuração*. Você pode encontrar sua chave em seu perfil no Weblate.

--config PATH

Substitui o caminho do arquivo de configuração, consulte *Arquivos de configuração*.

--config-section SECTION

Substitui a seção de arquivos de configuração em uso, consulte *Arquivos de configuração*.

Comandos

Os seguintes comandos estão disponíveis:

version

Imprime a versão atual.

list-languages

Lista os idiomas usados no Weblate.

list-projects

Lista os projetos no Weblate.

list-components

Lista os componentes no Weblate.

list-translations

Lista as traduções no Weblate.

show

Mostra o objeto do Weblate (tradução, componente ou projeto).

ls

Lista o objeto do Weblate (tradução, componente ou projeto).

commit

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

pull

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

push

Faz push das alterações do objeto Weblate para o repositório remoto (tradução, componente ou projeto).

reset

Novo na versão 0.7: Suportado desde o wlc 0.7.

Redefine as alterações no objeto Weblate para corresponder ao repositório remoto (tradução, componente ou projeto).

cleanup

Novo na versão 0.9: Suportado desde o wlc 0.9.

Remove quaisquer alterações não rastreadas em um objeto Weblate para corresponder ao repositório remoto (tradução, componente ou projeto).

repo

Exibe o status do repositório para um determinado objeto Weblate (tradução, componente ou projeto).

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 status do bloqueio.

lock

Novo na versão 0.5: Suportado desde o wlc 0.5.

Bloqueia o componente de tradução posterior no Weblate.

unlock

Novo na versão 0.5: Suportado desde o wlc 0.5.

Desbloqueia a tradução do componente Weblate.

changes

Novo na versão 0.7: Suportado desde o wlc 0.7 e o Weblate 2.10.

Exibe alterações para um determinado objeto.

download

Novo na versão 0.7: Suportado desde o wlc 0.7.

Baixa um arquivo de tradução.

--convert

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

--output

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

upload

Novo na versão 0.9: Suportado desde o wlc 0.9.

Baixa um arquivo de tradução.

--overwrite

Substitui as traduções existentes ao enviar.

--input

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

--method

Método de envio para usar, veja *Métodos de importação*.

--fuzzy

Processamento de textos aproximados (marcados para edição) (*vazio*, *process*, *approve*)

--author-name

Nome do autor, para substituir o usuário atualmente autenticado

--author-email

E-mail do autor, para substituir o usuário atualmente autenticado

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

1.13.6 Arquivos de configuração

.weblate, .weblate.ini, weblate.ini

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

Por arquivo de configuração de projeto

C:\Users\NAME\AppData\weblate.ini

Novo na versão 1.6.

Arquivo de configuração do usuário no Windows.

~/ .config/weblate

Arquivo de configuração do usuário

/etc/xdg/weblate

Arquivo de configuração para todo o sistema

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

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

key

Chave de API para acessar o Weblate.

url

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

translation

Caminho para a tradução padrão - componente ou projeto.

O arquivo de configuração é um arquivo INI, por exemplo:

```
[weblate]
url = https://hosted.weblate.org/api/
key = APIKEY
translation = weblate/application
```

Além disso, as chaves de API podem ser armazenadas na seção `[keys]`:

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

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

1.13.7 Exemplos

Imprimir a versão atual do programa:

```
$ wlc version
version: 0.1
```

Listar todos os projetos:

```
$ wlc list-projects
name: Hello
slug: hello
url: http://example.com/api/projects/hello/
web: https://weblate.org/
web_url: http://example.com/projects/hello/
```

Enviar arquivo de tradução:

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

Você também pode designar em qual projeto o `wlc` deve trabalhar:

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

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

(continua na próxima página)

(continuação da página anterior)

```
slug: application
template:
url: https://hosted.weblate.org/api/components/weblate/application/
vcs: git
web_url: https://hosted.weblate.org/projects/weblate/application/
```

Com esta configuração, é fácil fazer commit de alterações pendentes no projeto atual:

```
$ wlc commit
```

1.14 API Python do Weblate

1.14.1 Instalação

A API Python é distribuída separadamente, você precisa instalar o *Weblate Client* (wlc) para tê-la.

```
pip install wlc
```

1.14.2 wlc

WeblateException

exception `wlc.WeblateException`

Classe base para todas as exceções.

Weblate

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

Parâmetros

- **key** (*str*) – Chave do usuário
- **url** (*str*) – URL da API do servidor; se não especificado, o padrão é usado
- **config** (`wlc.config.WeblateConfig`) – Objeto de configuração, substitui quaisquer outros parâmetros.

Acessa a classe para a API, define a chave da API e, opcionalmente, a URL da API.

get (*path*)

Parâmetros

path (*str*) – Caminho da solicitação

Tipo de retorno

object

Executa uma única chamada de API GET.

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

Parâmetros

path (*str*) – Caminho da solicitação

Tipo de retorno

object

Executa uma única chamada de API GET.

1.14.3 `wlc.config`

`WeblateConfig`

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

Parâmetros

section (*str*) – Seção de configuração para uso

Analizador de arquivo de configuração seguindo a especificação XDG.

load (*path=None*)

Parâmetros

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

Carrega a configuração de um arquivo. Se nenhuma for especificada, carrega do arquivo de configuração `wlc (~/.config/wlc)` colocado em seu caminho de configuração XDG (`/etc/xdg/wlc`).

1.14.4 `wlc.main`

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

Parâmetros

- **settings** (*list*) – Configurações para substituir como lista de tuplas
- **stdout** (*object*) – objeto de arquivo stdout para saída de impressão, usa `sys.stdout` como padrão
- **args** (*list*) – Argumentos de linha de comando para processar, usa `sys.args` como padrão

Ponto de entrada principal para interface de linha de comando.

`@wlc.main.register_command (command)`

Decorador para registrar a classe `Command` no analisador principal usado por `main()`.

`Command`

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

Classe principal para invocação de comandos.

2.1 Instruções de configuração

2.1.1 Instalando o Weblate

Instalando usando Docker

Com a implantação do Weblate dockerizada, você pode colocar sua instância Weblate pessoal em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão.

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate em um único host (Weblate, banco de dados e servidor web):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de arquivos, banco de dados e Weblate).

Muitos usuários simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter seu tamanho mínimo fazendo clones rasos.

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

Instalação

Os exemplos a seguir presumem que você tem um ambiente Docker funcional, com `docker-compose` instalado. Verifique a documentação do Docker para obter instruções.

1. Clone o repositório `weblate-docker`:

```
git clone https://github.com/WeblateOrg/docker-compose.git weblate-docker
cd weblate-docker
```

2. Crie um arquivo `docker-compose.override.yml` com suas configurações. Veja [Variáveis de ambiente do Docker](#) para uma lista completa das variáveis de ambiente.

```
version: '3'
services:
  weblate:
    ports:
      - 80:8080
    environment:
      WEBLATE_EMAIL_HOST: smtp.example.com
      WEBLATE_EMAIL_HOST_USER: user
      WEBLATE_EMAIL_HOST_PASSWORD: pass
      WEBLATE_SERVER_EMAIL: weblate@example.com
      WEBLATE_DEFAULT_FROM_EMAIL: weblate@example.com
      WEBLATE_SITE_DOMAIN: weblate.example.com
      WEBLATE_ADMIN_PASSWORD: password for the admin user
      WEBLATE_ADMIN_EMAIL: weblate.admin@example.com
```

Nota: Se `WEBLATE_ADMIN_PASSWORD` não estiver definida, o usuário `admin` é criado com uma senha aleatória mostrada na primeira inicialização.

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

3. Inicie os contêineres do Weblate:

```
docker-compose up
```

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

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

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

Ver também:

Invocando comandos de gerenciamento

Escolhendo a tag do hub Docker

Você 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 em um ambiente de produção
<VERSION>-<	Versão estável do Weblate	Implantação bem definida em um ambiente de produção
edge	Lançamento estável do Weblate com alterações de desenvolvimento no contêiner Docker (por exemplo, dependências atualizadas)	Atualizações contínuas em um ambiente de teste
edge-<DATE>	Lançamento estável do Weblate com alterações de desenvolvimento no contêiner Docker (por exemplo, dependências atualizadas)	Implantação bem definida em um ambiente de teste
bleeding	Versão de desenvolvimento do Weblate do Git	Atualizações contínuas para testar os próximos recursos do Weblate
bleeding-<I	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.

Contêiner Docker com suporte a HTTPS

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

Usando seus próprios certificados SSL

Novo na versão 3.8-3.

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

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

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

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

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

Se você já hospeda outros sites no mesmo servidor, é provável que as portas 80 e 443 sejam usadas por um proxy reverso, como NGINX. Para passar a conexão HTTPS do NGINX para o contêiner do docker, você pode usar a seguinte configuração:


```
server {
    listen 443;
    listen [::]:443;

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

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

Substitua <SITE_URL>, <SITE> e <EXPOSED_DOCKER_PORT> por valores reais de seu ambiente.

Certificados SSL automáticos usando Let's Encrypt

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

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

Sempre que invocar **docker-compose**, você precisa passar os dois arquivos para ele, e então fazer:

```
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml
↪build
docker-compose -f docker-compose-https.yml -f docker-compose-https.override.yml up
```

Atualizando o contêiner Docker

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

Alterado na versão 4.10-1: Desde o Weblate 4.10-1, o contêiner Docker usa Django 4.0, o que requer PostgreSQL 10 ou mais recente, atualize-o antes de atualizar o Weblate. Veja [Atualizar da 4.9 para 4.10](#) e [Atualizando contêiner PostgreSQL](#).

Você pode fazer isso mantendo 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: Atualizações entre versões principais não são suportadas pelo Weblate. Se você estiver na série 3.x e quiser atualizar para 4.x, primeiro atualize para a imagem 4.0.x-y mais recente (no momento em que escrevo esta é a 4.0.4-5), que faça a migração e, em seguida, continue atualizando para as versões mais recentes.

Você também pode querer atualizar o repositório `docker-compose`, embora não seja necessário na maioria dos casos. Veja [Atualizando contêiner PostgreSQL](#) para atualizar o servidor PostgreSQL.

Atualizando contêiner PostgreSQL

Os contêineres PostgreSQL não oferecem suporte a atualização automática entre versões, você precisa realizar a atualização manualmente. Os passos a seguir mostram uma das opções de atualização.

Ver também:

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

1. Pare o contêiner do Weblate:

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

2. Faça backup do banco de dados:

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

3. Pare o contêiner de banco de dados:

```
docker-compose stop database
```

4. Remova o volume do PostgreSQL:

```
docker-compose rm -v database
docker volume remove weblate-docker_postgres-data
```

5. Ajuste o `docker-compose.yml` para usar a nova versão do PostgreSQL.

6. Inicie o contêiner de banco de dados:

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

7. Restaure o banco de dados a partir do backup:

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

8. (Optional) Update password for the Weblate user. This might be needed when migrating to PostgreSQL 14 or 15 as way of storing passwords has been changed:

```
docker-compose exec -T database psql --username weblate --dbname postgres -c
↳ "ALTER USER weblate WITH PASSWORD 'weblate'"
```

9. Inicie todos os contêineres restantes:

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

Autenticação como administrador

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

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

Ver também:

`WEBLATE_ADMIN_PASSWORD`, `WEBLATE_ADMIN_NAME`, `WEBLATE_ADMIN_EMAIL`

Número de processos e consumo de memória

O número de processos de trabalho para uWSGI e Celery é determinado automaticamente com base no número de CPUs. Isso funciona bem para a maioria das máquinas virtuais em nuvem, pois normalmente têm poucas CPUs e boa quantidade de memória.

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

```
environment:
  WEBLATE_WORKERS: 2
```

Você 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
```

Ver também:

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

Dimensionamento horizontal

Novo na versão 4.6.

Você pode executar vários contêineres Weblate para dimensionar o serviço horizontalmente. O volume `/app/data` deve ser compartilhado por todos os contêineres, é recomendado usar um sistema de arquivos de cluster como o GlusterFS para isso. O volume `/app/cache` deve ser separado para cada contêiner.

Cada contêiner Weblate tem um papel definido usando 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.

Você 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 contêiner Docker usando as variáveis de ambiente descritas abaixo.

Se você precisar definir uma configuração não exposta por meio de variáveis de ambiente do Docker, consulte *Configuração além das variáveis de ambiente*.

Configurações genéricas

WEBLATE_DEBUG

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

Exemplo:

```
environment:
  WEBLATE_DEBUG: 1
```

Ver também:

Desabilitar o modo de depuração

WEBLATE_LOGLEVEL

Configura o detalhamento do log.

WEBLATE_LOGLEVEL_DATABASE

Configura o log da verbosidade das consultas ao banco de dados.

WEBLATE_SITE_TITLE

Altera o título do site mostrado no cabeçalho de todas as páginas.

WEBLATE_SITE_DOMAIN

Configura o domínio do site. Este parâmetro é obrigatório.

Ver também:

Definir domínio correto do site, SITE_DOMAIN

WEBLATE_ADMIN_NAME

WEBLATE_ADMIN_EMAIL

Configura o nome e o e-mail do administrador do site. É usado para *ADMINS* e para criar o usuário *admin* (veja *WEBLATE_ADMIN_PASSWORD* para mais informações).

Exemplo:

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

Ver também:

Autenticação como administrador, Configurar corretamente administradores, ADMINS

WEBLATE_ADMIN_PASSWORD

Define a senha para o usuário *admin*.

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

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

Ver também:

Autenticação como administrador, WEBLATE_ADMIN_PASSWORD, WEBLATE_ADMIN_PASSWORD_FILE, WEBLATE_ADMIN_NAME, WEBLATE_ADMIN_EMAIL

WEBLATE_ADMIN_PASSWORD_FILE

Define o caminho para um arquivo contendo a senha para o usuário *admin*.

Ver também:

WEBLATE_ADMIN_PASSWORD

WEBLATE_SERVER_EMAIL

O endereço de e-mail a partir do qual mensagens de erro são enviadas.

Ver também:

SERVER_EMAIL, Configurar envio de e-mail

WEBLATE_DEFAULT_FROM_EMAIL

Configura o endereço para e-mails de saída.

Ver 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 usando *ALLOWED_HOSTS*.

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

Exemplo:

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

Ver também:

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

WEBLATE_REGISTRATION_OPEN

Configura se os registros são abertos alternando *REGISTRATION_OPEN*.

Exemplo:

```
environment:
  WEBLATE_REGISTRATION_OPEN: 0
```

WEBLATE_REGISTRATION_ALLOW_BACKENDS

Configura quais métodos de autenticação podem ser usados para criar uma nova conta via *REGISTRATION_ALLOW_BACKENDS*.

Exemplo:

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

WEBLATE_REGISTRATION_REBIND

Novo na versão 4.16.

Configures `REGISTRATION_REBIND`.

WEBLATE_TIME_ZONE

Configura o fuso horário usado no Weblate, veja `django: TIME_ZONE`.

Nota: Para alterar o fuso horário do próprio contêiner do Docker, use a variável de ambiente `TZ`.

Exemplo:

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

WEBLATE_ENABLE_HTTPS

Faz com que o Weblate presuma que é operado por trás de um proxy HTTPS reverso, faz com que o Weblate use HTTPS em e-mail e links de API ou defina marcadores seguros em cookies.

Dica: Por favor, consulte a documentação de `ENABLE_HTTPS` para possíveis advertências.

Nota: Isso não faz com que o contêiner Weblate aceite conexões HTTPS, você precisa configurar isso também, consulte *Contêiner Docker com suporte a HTTPS* para exemplos.

Exemplo:

```
environment:
  WEBLATE_ENABLE_HTTPS: 1
```

Ver também:

`ENABLE_HTTPS` Definir domínio correto do site, `WEBLATE_SECURE_PROXY_SSL_HEADER`

WEBLATE_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.

Ver 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 contêiner Weblate.

Habilita `IP_BEHIND_REVERSE_PROXY` e define `IP_PROXY_HEADER`.

Nota: O formato deve estar de acordo com as expectativas do Django. O Django *transforma* nomes de cabeçalho HTTP brutos da seguinte forma:

- converte todos os caracteres em maiúsculas
- substitui todos os hifenes por sublinhados
- prefixa o prefixo `HTTP_`

Portanto, X-Forwarded-For seria mapeado para HTTP_X_FORWARDED_FOR.

Exemplo:

```
environment:
  WEBLATE_IP_PROXY_HEADER: HTTP_X_FORWARDED_FOR
```

WEBLATE_SECURE_PROXY_SSL_HEADER

Uma tupla que representa uma combinação de cabeçalho/valor HTTP que significa que uma solicitação é segura. Isso é necessário quando o Weblate está sendo executado por trás de um proxy reverso fazendo a terminação SSL que não passa cabeçalhos HTTPS padrão.

Exemplo:

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

Ver também:

[SECURE_PROXY_SSL_HEADER](#)

WEBLATE_REQUIRE_LOGIN

Habilita [REQUIRE_LOGIN](#) para impor autenticação em todo o Weblate.

Exemplo:

```
environment:
  WEBLATE_REQUIRE_LOGIN: 1
```

WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_ADD_LOGIN_REQUIRED_URLS_EXCEPTIONS

WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS

Adiciona exceções de URL para autenticação necessária para toda a instalação do Weblate usando [LOGIN_REQUIRED_URLS_EXCEPTIONS](#).

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

WEBLATE_GOOGLE_ANALYTICS_ID

Configura o ID para o Google Analytics alterando [GOOGLE_ANALYTICS_ID](#).

WEBLATE_GITHUB_USERNAME

WEBLATE_GITHUB_TOKEN

WEBLATE_GITHUB_HOST

Configures GitHub pull-requests integration by changing [GITHUB_CREDENTIALS](#).

Ver também:

[Pull requests do GitHub](#)

WEBLATE_GITLAB_USERNAME

WEBLATE_GITLAB_TOKEN

WEBLATE_GITLAB_HOST

Configures GitLab merge-requests integration by changing [GITLAB_CREDENTIALS](#).

Ver também:

[Merge requests do GitLab](#)

WEBLATE_GITEA_USERNAME

WEBLATE_GITEA_TOKEN

WEBLATE_GITEA_HOST

Configures Gitea pull-requests integration by changing *GITEA_CREDENTIALS*.

Ver também:

Pull requests do Gitea

WEBLATE_PAGURE_USERNAME

WEBLATE_PAGURE_TOKEN

WEBLATE_PAGURE_HOST

Configures Pagure merge-requests integration by changing *PAGURE_CREDENTIALS*.

Ver também:

Merge requests do Pagure

WEBLATE_BITBUCKETSERVER_USERNAME

WEBLATE_BITBUCKETSERVER_TOKEN

WEBLATE_BITBUCKETSERVER_HOST

Configures Bitbucket Server pull-requests integration by changing *BITBUCKETSERVER_CREDENTIALS*.

Ver também:

Bitbucket Server pull requests

WEBLATE_DEFAULT_PULL_MESSAGE

Configura o título e a mensagem padrão para pull requests via API alterando *DEFAULT_PULL_MESSAGE*

Ver também:

DEFAULT_PULL_MESSAGE

WEBLATE_SIMPLIFY_LANGUAGES

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

WEBLATE_DEFAULT_ACCESS_CONTROL

Configura o padrão *Controle de acesso* para novos projetos, veja *DEFAULT_ACCESS_CONTROL*.

WEBLATE_DEFAULT_RESTRICTED_COMPONENT

Configura o valor padrão para *Acesso restrito* para novos componentes, veja *DEFAULT_RESTRICTED_COMPONENT*.

WEBLATE_DEFAULT_TRANSLATION_PROPAGATION

Configura o valor padrão para *Permitir propagação de tradução* para novos componentes, veja *DEFAULT_TRANSLATION_PROPAGATION*.

WEBLATE_DEFAULT_COMMITER_EMAIL

Configura *DEFAULT_COMMITER_EMAIL*.

WEBLATE_DEFAULT_COMMITER_NAME

Configura *DEFAULT_COMMITER_NAME*.

WEBLATE_DEFAULT_SHARED_TM

Configura *DEFAULT_SHARED_TM*.

WEBLATE_AKISMET_API_KEY

Configura a chave API do Akismet, veja *AKISMET_API_KEY*.

WEBLATE_GPG_IDENTITY

Configura a assinatura GPG de commits, veja [WEBLATE_GPG_IDENTITY](#).

Ver também:

Assinando commits do Git com GnuPG

WEBLATE_URL_PREFIX

Configura o prefixo da URL onde o Weblate está sendo executado, veja [URL_PREFIX](#).

WEBLATE_SILENCED_SYSTEM_CHECKS

Configura verificações que você não deseja que sejam mostradas, veja `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.

Ver também:

Política de segurança de conteúdo, [CSP_SCRIPT_SRC](#), [CSP_IMG_SRC](#), [CSP_CONNECT_SRC](#), [CSP_STYLE_SRC](#), [CSP_FONT_SRC](#)

WEBLATE_LICENSE_FILTER

Configura [LICENSE_FILTER](#).

WEBLATE_LICENSE_REQUIRED

Configura [LICENSE_REQUIRED](#)

WEBLATE_WEBSITE_REQUIRED

Configura [WEBSITE_REQUIRED](#)

WEBLATE_HIDE_VERSION

Configura [HIDE_VERSION](#).

WEBLATE_BASIC_LANGUAGES

Configura [BASIC_LANGUAGES](#).

WEBLATE_DEFAULT_AUTO_WATCH

Configura [DEFAULT_AUTO_WATCH](#).

WEBLATE_RATELIMIT_ATTEMPTS**WEBLATE_RATELIMIT_LOCKOUT****WEBLATE_RATELIMIT_WINDOW**

Novo na versão 4.6.

Configura o limitador de taxa.

Dica: Você 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](#).

Ver também:

Limitação de taxa, [RATELIMIT_ATTEMPTS](#), [RATELIMIT_WINDOW](#), [RATELIMIT_LOCKOUT](#)

WEBLATE_API_RATELIMIT_ANON

WEBLATE_API_RATELIMIT_USER

Novo na versão 4.11.

Configura a limitação de taxa da API. O padrão é 100/day para usuários anônimos e 5000/hour para usuários autenticados.

Ver também:

Limitação de taxa da API

WEBLATE_ENABLE_HOOKS

Novo na versão 4.13.

Configura *ENABLE_HOOKS*.

WEBLATE_ENABLE_AVATARS

Novo na versão 4.6.1.

Configura *ENABLE_AVATARS*.

WEBLATE_AVATAR_URL_PREFIX

Novo na versão 4.15.

Configura *AVATAR_URL_PREFIX*.

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.

Configura *ENABLE_SHARING*.

WEBLATE_EXTRA_HTML_HEAD

Novo na versão 4.15.

Configura *EXTRA_HTML_HEAD*.

WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE

Novo na versão 4.15.

Configura *PRIVATE_COMMIT_EMAIL_TEMPLATE*.

WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN

Novo na versão 4.15.

Configura *PRIVATE_COMMIT_EMAIL_OPT_IN*.

WEBLATE_CORS_ALLOWED_ORIGINS

Novo na versão 4.16.

Allow CORS requests from given origins.

Exemplo:

```
environment:
  WEBLATE_CORS_ALLOWED_ORIGINS: https://example.com,https://weblate.org
```

Configurações de sugestões automáticas

Alterado na versão 4.13: Serviços de sugestões automáticas agora são configurados na interface de usuário, consulte *Configurando sugestões automáticas*.

As variáveis de ambiente atuais são importadas durante a migração para o 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

Caminho para o arquivo que contém a senha de ligação do servidor LDAP.

Ver 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)
```

Ver 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

Habilita *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

Habilita *Autenticação por Bitbucket*.

Facebook

`WEBLATE_SOCIAL_AUTH_FACEBOOK_KEY`

`WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET`

Habilita *OAuth 2 do Facebook*.

Google

`WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY`

`WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET`

`WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS`

`WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS`

Habilita *Google OAuth 2*.

GitLab

`WEBLATE_SOCIAL_AUTH_GITLAB_KEY`

`WEBLATE_SOCIAL_AUTH_GITLAB_SECRET`

`WEBLATE_SOCIAL_AUTH_GITLAB_API_URL`

Habilita *OAuth 2 do GitLab*.

Gitea

`WEBLATE_SOCIAL_AUTH_GITEA_API_URL`

`WEBLATE_SOCIAL_AUTH_GITEA_KEY`

`WEBLATE_SOCIAL_AUTH_GITEA_SECRET`

Habilita autenticação por Gitea.

Active Directory do Azure

`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY`

`WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET`

Habilita autenticação por Active Directory do Azure, veja *Active Directory do Microsoft Azure*.

Active Directory do Azure com suporte a Tenant

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY`

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET`

`WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID`

Habilita autenticação por Active Directory do Azure com suporte a Tenant, veja *Active Directory do Microsoft Azure*.

Keycloak

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY`

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET`

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY`

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM`

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL`

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL`

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE`

`WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE`

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

Fornecedores Linux

Você pode habilitar a autenticação usando serviços de autenticação de fornecedores Linux, definindo as seguintes variáveis para qualquer valor.

`WEBLATE_SOCIAL_AUTH_FEDORA`

`WEBLATE_SOCIAL_AUTH_OPENSUSE`

`WEBLATE_SOCIAL_AUTH_UBUNTU`

Slack

`WEBLATE_SOCIAL_AUTH_SLACK_KEY`

`SOCIAL_AUTH_SLACK_SECRET`

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

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`

Configura a integração genérica do OpenID Connect.

Ver também:

[OIDC \(OpenID Connect\)](#)

SAML

Chaves SAML autoassinadas são geradas automaticamente na primeira inicialização do contêiner. Caso você queira usar chaves próprias, coloque o certificado e a chave privada em `/app/data/ssl/saml.crt` e `/app/data/ssl/saml.key`.

WEBLATE_SAML_IDP_ENTITY_ID

WEBLATE_SAML_IDP_URL

WEBLATE_SAML_IDP_X509CERT

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

Desabilita autenticação por e-mail quando definido com algum valor. Veja [Desativando autenticação por senha](#).

Configuração de banco de dados PostgreSQL

O banco de dados é criado por `docker-compose.yml`, então essas configurações afetam os contêineres Weblate e PostgreSQL.

Ver também:

Configuração de banco de dados para o Weblate

POSTGRES_PASSWORD

Senha do PostgreSQL.

POSTGRES_PASSWORD_FILE

Caminho para o arquivo que contém a senha do PostgreSQL. Use como uma alternativa para `POSTGRES_PASSWORD`.

POSTGRES_USER

Nome de usuário do PostgreSQL.

POSTGRES_DATABASE

Nome do banco de dados PostgreSQL.

POSTGRES_HOST

Nome de host ou endereço IP do servidor PostgreSQL. O padrão é `database`.

POSTGRES_PORT

Porta do servidor PostgreSQL. O padrão é nenhum (usa o valor padrão).

POSTGRES_SSL_MODE

Configura como o PostgreSQL lida com SSL em conexão com o servidor, para as opções possíveis, consulte [SSL Mode Descriptions](#)

POSTGRES_ALTER_ROLE

Configura o nome da função para alterar durante as migrações, consulte [Configurando Weblate para usar PostgreSQL](#).

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 padrão).

Habilitar a persistência da conexão normalmente causará uma conexão mais aberta com o banco de dados. Por favor, ajuste sua configuração do banco de dados antes de habilitar.

Exemplo de configuração:

```
environment:
  POSTGRES_CONN_MAX_AGE: 3600
```

Ver também:

[CONN_MAX_AGE](#), [Conexões persistentes](#)

POSTGRES_DISABLE_SERVER_SIDE_CURSORS

Novo na versão 4.9.1.

Desabilita 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
```

Ver também:

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

Configurações de backup de banco de dados

Ver também:

Dados despejados para os backups

WEBLATE_DATABASE_BACKUP

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

Configuração do servidor de cache

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

Ver também:

Habilitar o cache

REDIS_HOST

O nome de host ou endereço IP do servidor Redis. O padrão é `cache`.

REDIS_PORT

A porta do servidor Redis. O padrão é `6379`.

REDIS_DB

O número do banco de dados Redis, o padrão é `1`.

REDIS_PASSWORD

A senha do servidor Redis, não usada por padrão.

REDIS_PASSWORD_FILE

Caminho para o arquivo que contém a senha do servidor Redis.

Ver também:

REDIS_PASSWORD

REDIS_TLS

Habilita o uso de SSL para conexão Redis.

REDIS_VERIFY_SSL

Pode ser usado para desativar a verificação de certificado SSL para conexão Redis.

Configuração do servidor de e-mail

Para fazer com que o e-mail de saída funcione, você precisa fornecer um servidor de e-mail.

Exemplo de configuração TLS:

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

Exemplo de configuração SSL:

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

Ver também:

Configuração de e-mail de saída

WEBLATE_EMAIL_HOST

Nome de host ou endereço IP do servidor de correio.

Ver também:

WEBLATE_EMAIL_PORT, WEBLATE_EMAIL_USE_SSL, WEBLATE_EMAIL_USE_TLS, EMAIL_HOST

WEBLATE_EMAIL_PORT

Porta do servidor de correio, o padrão é 25.

Ver também:

EMAIL_PORT

WEBLATE_EMAIL_HOST_USER

Usuário da autenticação por e-mail.

Ver também:

EMAIL_HOST_USER

WEBLATE_EMAIL_HOST_PASSWORD

Senha da autenticação por e-mail.

Ver também:

EMAIL_HOST_PASSWORD

WEBLATE_EMAIL_HOST_PASSWORD_FILE

Caminho para o arquivo contendo a senha da autenticação por e-mail.

Ver 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 você estiver tendo problemas, consulte a configuração TLS explícita [*WEBLATE_EMAIL_USE_TLS*](#).

Alterado na versão 4.11: O suporte a SSL/TLS é habilitado automaticamente com base em [*WEBLATE_EMAIL_PORT*](#).

Ver também:

[*WEBLATE_EMAIL_PORT*](#), [*WEBLATE_EMAIL_USE_TLS*](#), [*EMAIL_USE_SSL*](#)

WEBLATE_EMAIL_USE_TLS

Se deve usar uma conexão TLS (segura) ao falar com o servidor SMTP. Isso é usado para conexões TLS explícitas, geralmente na porta 587 ou 25. Se você estiver tendo conexões travadas, consulte a configuração TLS implícita [*WEBLATE_EMAIL_USE_SSL*](#).

Alterado na versão 4.11: O suporte a SSL/TLS é habilitado automaticamente com base em [*WEBLATE_EMAIL_PORT*](#).

Ver também:

[*WEBLATE_EMAIL_PORT*](#), [*WEBLATE_EMAIL_USE_SSL*](#), [*EMAIL_USE_TLS*](#)

WEBLATE_EMAIL_BACKEND

Configura o back-end do Django para usar no envio de e-mails.

Ver também:

[*Configurar envio de e-mail*](#), [*EMAIL_BACKEND*](#)

WEBLATE_AUTO_UPDATE

Configura se e como o Weblate deve atualizar os repositórios.

Ver 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 habilitar o suporte para Rollbar, defina o seguinte:

ROLLBAR_KEY

Seu token de acesso ao servidor de postagem Rollbar.

ROLLBAR_ENVIRONMENT

Seu ambiente Rollbar, o padrão é `production`.

Para habilitar o suporte para Sentry, defina o seguinte:

SENTRY_DSN

Seu DSN no Sentry.

SENTRY_ENVIRONMENT

Seu ambiente no Sentry (opcional).

CDN de localização

WEBLATE_LOCALIZE_CDN_URL

WEBLATE_LOCALIZE_CDN_PATH

Novo na versão 4.2.1.

Configuração para *CDN de localização do JavaScript*.

O `WEBLATE_LOCALIZE_CDN_PATH` é o caminho dentro do contêiner. Ele deve ser armazenado no volume persistente e não no armazenamento temporário.

Uma das possibilidades é armazenar isso dentro do diretório de dados do Weblate:

```
environment:
  WEBLATE_LOCALIZE_CDN_URL: https://cdn.example.com/
  WEBLATE_LOCALIZE_CDN_PATH: /app/data/l10n-cdn
```

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

Ver também:

weblate-cdn, `LOCALIZE_CDN_URL`, `LOCALIZE_CDN_PATH`

Alterando aplicativos, verificações, extensões ou correções automáticas habilitados

Novo na versão 3.8-5.

A configuração embutida de verificações, extensões ou correções automática habilitados pode ser ajustada pelas seguintes variáveis:

WEBLATE_ADD_APPS

WEBLATE_REMOVE_APPS

WEBLATE_ADD_CHECK

WEBLATE_REMOVE_CHECK

WEBLATE_ADD_AUTOFIX

WEBLATE_REMOVE_AUTOFIX

WEBLATE_ADD_ADDONS

WEBLATE_REMOVE_ADDONS

Exemplo:

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

Ver também:

[CHECK_LIST](#), [AUTOFIX_LIST](#), [WEBLATE_ADDONS](#), [INSTALLED_APPS](#)

Configurações do contêiner

WEBLATE_WORKERS

Novo na versão 4.6.1.

Número base de processos de trabalho em execução no contêiner. Quando não definido, é determinado automaticamente na inicialização do contêiner com base no número 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](#). Você 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 você 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, o número de workers simultâneos é baseado em [WEBLATE_WORKERS](#).

Exemplo:

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

Ver também:

[Opções de worker do Celery](#), *Tarefas de fundo usando 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 contêiner. Use isto para *Dimensionamento horizontal*.

Os seguintes serviços são definidos:

celery-beat

Agendador de tarefas do Celery, apenas uma instância deve estar em execução. Este contêiner 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 contêiner Docker

Há dois volumes (dados e cache) exportados pelo contêiner Weblate. Os outros contêineres de serviço (PostgreSQL ou Redis) também têm seus volumes de dados, 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, contêiner e nomes de volume). No contêiner, ele é montado como `/app/data`.

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

Ao criar os volumes manualmente, os diretórios devem pertencer ao UID 1000, pois é o usuário usado dentro do contêiner.

Ver também:

[Documentação sobre volumes do Docker](#)

Configuração além das variáveis de ambiente

As *variáveis de ambiente do Docker* destinam-se a expor a maioria das *definições de configuração* de relevância para as instalações do Weblate.

Se você encontrar uma configuração que não está exposta como uma variável de ambiente, e você acredita que deveria estar, sintá-se à vontade para *pedir que ela seja exposta em uma versão futura do Weblate*.

Se você precisar modificar uma configuração que não está exposta como uma variável de ambiente do Docker, ainda poderá fazê-lo *a partir do volume de dados* ou estendendo a imagem do Docker.

Ver também:

Personalizando o Weblate

Substituindo as configurações do volume de dados

Você pode criar um arquivo em `/app/data/settings-override.py`, ou seja, na raiz do *volume de dados*, para estender ou substituir as configurações definidas por meio de variáveis de ambiente.

Substituindo as configurações estendendo a imagem do Docker

Para substituir as configurações no nível da imagem do Docker em vez do volume de dados:

1. *Crie um pacote Python personalizado.*
2. Adicione um módulo ao seu pacote que importe todas as configurações de `weblate.settings_docker`.

Por exemplo, dentro da estrutura de pacote de exemplo definida em *Criando um módulo Python*, você pode criar um arquivo em `weblate_customization/weblate_customization/settings.py` com o seguinte código inicial:

```
from weblate.settings_docker import *
```

3. Crie um `Dockerfile` personalizado que herde da imagem oficial do Weblate Docker e, em seguida, instale seu pacote e aponte a variável de ambiente `DJANGO_SETTINGS_MODULE` para seu módulo de configurações:

```
FROM weblate/weblate

USER root

COPY weblate_customization /usr/src/weblate_customization
RUN pip install --no-cache-dir /usr/src/weblate_customization
ENV DJANGO_SETTINGS_MODULE=weblate_customization.settings

USER 1000
```

4. Em vez de usar a imagem Docker oficial do Weblate, construa uma imagem personalizada a partir deste arquivo `Dockerfile`.

Não existe nenhuma maneira limpa <<https://github.com/docker/compose/issues/7231>> de fazer isso com `docker-compose.override.yml`. Você *poderia* adicionar `build: .` ao nó `weblate` nesse arquivo, mas sua imagem personalizada será marcada como `weblate/weblate` em seu sistema, o que pode ser problemático.

Então, ao invés de usar o `docker-compose.yml` direto do *repositório oficial*, não modificado, e estendê-lo através do `docker-compose.override.yml`, você pode querer fazer uma cópia do arquivo `docker-compose.yml` oficial, e editar sua cópia para substituir `image: weblate/weblate` por `build: ..`

Consulte a *Referência de compilação do arquivo Compose* para obter detalhes sobre como criar imagens a partir da fonte ao usar `docker-compose`.

5. Estenda seu módulo de configurações personalizadas para definir ou redefinir as configurações.

Você pode definir as configurações antes ou depois da instrução de importação acima para determinar quais configurações têm precedência. As configurações definidas antes da instrução de importação podem ser substituídas por variáveis de ambiente e substituições de configuração definidas no volume de dados. A configuração definida após a instrução de importação não pode ser substituída.

Você também pode ir mais longe. Por exemplo, você pode reproduzir algumas das coisas que `weblate.docker_settings` faz, como expor configurações como variáveis de ambiente ou permitir a substituição de configurações de arquivos Python no volume de dados.

Substituindo o logotipo e outros arquivos estáticos

Novo na versão 3.8-5.

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

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

Essa abordagem também pode ser usada para substituir os modelos Weblate. Por exemplo documentos *legais* podem ser colocados em `/app/data/python/customize/templates/legal/documents`.

Como alternativa, você também pode incluir o próprio módulo (veja `../customize`) e adicioná-lo como um volume separado ao contêiner do Docker, por exemplo:

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

Configurando o servidor PostgreSQL

O contêiner PostgreSQL usa a configuração padrão do PostgreSQL e não utilizará efetivamente seus núcleos de CPU ou memória. 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/>.

Partes internas do contêiner

O contêiner está usando **supervisor** para iniciar serviços individuais. No caso de *Dimensionamento horizontal*, ele inicia apenas um único serviço em um contêiner.

Para verificar o status dos serviços, use:

```
docker-compose exec --user weblate weblate supervisorctl status
```

Existem serviços individuais para cada fila Celery (veja *Tarefas de fundo usando Celery* para detalhes). Você pode interromper o processamento de algumas tarefas parando o worker apropriado:

```
docker-compose exec --user weblate weblate supervisorctl stop celery-translate
```

Instalando no Debian e Ubuntu

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate em um único host (Weblate, banco de dados e servidor web):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de arquivos, banco de dados e Weblate).

Muitos usuários simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter seu tamanho mínimo fazendo clones rasos.

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

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
apt install -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 libjpeg-dev build-essential \
  python3-gdbm python3-dev python3-pip python3-virtualenv virtualenv git
```

Instale as dependências opcionais desejadas, dependendo dos recursos que você pretende usar (veja *Dependências opcionais*):

```
apt install -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 *Executando servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo usando Celery*. Dependendo do tamanho da sua instalação, você pode querer executar esses componentes em servidores dedicados.

As instruções de instalação local:

```
# Web server option 1: NGINX and uWSGI
apt install -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
```

(continua na próxima página)

(continuação da página anterior)

```
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 usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você não estiver familiarizado com ele, consulte [User Guide](#) do virtualenv.

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, você precisa reconstruir o pacote em seu sistema:

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

Configurando o Weblate

Nota: O seguinte assume que o virtualenv utilizado pela Weblate é ativado (executando `. ~/weblate-env/bin/bin/activate`). Caso contrário, especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas você vai querer mais mudanças para configuração de produção, veja [Ajustando a configuração](#).

3. Crie o banco de dados e sua estrutura para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para uma configuração pronta para produção):

```
weblate migrate
```

4. Crie uma conta para o usuário administrador e copie sua senha para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

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

```
weblate collectstatic
```

6. Compacte os arquivos JavaScript e CSS (opcional, veja [Comprimindo os ativos do cliente](#)):

```
weblate compress
```

7. Inicie os workers do Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. [Tarefas de fundo usando Celery](#) tem mais informações:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

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

```
weblate runserver
```

Após a instalação

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

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Faça o login com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` (veja [Interface de gerenciamento](#)) ou usando **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que você precisa especificar aqui é o nome do projeto e seu site.

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

Os campos importantes aqui são: *Nome do componente*, *Repositório do código-fonte* e *Máscara de arquivo* para localizar arquivos traduzíveis. Weblate tem suporte a uma ampla gama de formatos, incluindo *GNU gettext*, *Recurso de textos de Android*, *Textos de Apple iOS*, *Propriedades Java*, *Formato stringsdict* ou *Formato Fluent*, consulte [Formatos de arquivos suportados](#) para mais detalhes.

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

Instalando no SUSE e openSUSE

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate em um único host (Weblate, banco de dados e servidor web):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de arquivos, banco de dados e Weblate).

Muitos usuários simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter seu tamanho mínimo fazendo clones rasos.

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

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
zypper install \
  libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
  libyaml-devel libffi-devel cairo-devel pango-devel \
  gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
  python3-devel git
```

Instale as dependências opcionais desejadas, dependendo dos recursos que você pretende usar (veja *Dependências opcionais*):

```
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 *Executando servidor*, *Configuração de banco de dados para o Weblate*, *Tarefas de fundo usando Celery*. Dependendo do tamanho da sua instalação, você pode querer executar esses componentes em servidores dedicados.

As instruções de instalação local:

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

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

# Caching backend: Redis
zypper install redis-server
```

(continua na próxima página)

(continuação da página anterior)

```
# Database server: PostgreSQL
zypper install postgresql postgresql-contrib

# SMTP server
zypper install postfix
```

Módulos do Python

Dica: Estamos usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você não estiver familiarizado com ele, consulte [User Guide](#) do virtualenv.

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, você precisa reconstruir o pacote em seu sistema:

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

Configurando o Weblate

Nota: O seguinte assume que o virtualenv utilizado pela Weblate é ativado (executando `. ~/weblate-env/bin/bin/activate`). Caso contrário, especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas você vai querer mais mudanças para configuração de produção, veja [Ajustando a configuração](#).

3. Crie o banco de dados e sua estrutura para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para uma configuração pronta para produção):

```
weblate migrate
```

4. Crie uma conta para o usuário administrador e copie sua senha para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

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

```
weblate collectstatic
```

6. Compacte os arquivos JavaScript e CSS (opcional, veja [Comprindo os ativos do cliente](#)):

```
weblate compress
```

7. Inicie os workers do Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. [Tarefas de fundo usando Celery](#) tem mais informações:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

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

```
weblate runserver
```

Após a instalação

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

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Faça o login com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` (veja [Interface de gerenciamento](#)) ou usando **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que você precisa especificar aqui é o nome do projeto e seu site.

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

Os campos importantes aqui são: *Nome do componente*, *Repositório do código-fonte* e *Máscara de arquivo* para localizar arquivos traduzíveis. Weblate tem suporte a uma ampla gama de formatos, incluindo *GNU gettext*, *Recurso de textos de Android*, *Textos de Apple iOS*, *Propriedades Java*, *Formato stringsdict* ou *Formato Fluent*, consulte [Formatos de arquivos suportados](#) para mais detalhes.

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

Instalando no RedHat, Fedora e CentOS

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate em um único host (Weblate, banco de dados e servidor web):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de arquivos, banco de dados e Weblate).

Muitos usuários simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter seu tamanho mínimo fazendo clones rasos.

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

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja [Requisitos de software](#)):

```
dnf install \
    libxslt-devel libxml2-devel freetype-devel libjpeg-devel zlib-devel \
    libyaml-devel libffi-devel cairo-devel pango-devel \
    gobject-introspection-devel libacl-devel python3-pip python3-virtualenv \
    python3-devel git
```

Instale as dependências opcionais desejadas, dependendo dos recursos que você pretende usar (veja [Dependências opcionais](#)):

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

Opcionalmente, instale o software para executar o servidor de produção, veja [Executando servidor](#), [Configuração de banco de dados para o Weblate](#), [Tarefas de fundo usando Celery](#). Dependendo do tamanho da sua instalação, você pode querer executar esses componentes em servidores dedicados.

As instruções de instalação local:

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

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

# Caching backend: Redis
dnf install redis
```

(continua na próxima página)

(continuação da página anterior)

```
# Database server: PostgreSQL
dnf install postgresql postgresql-contrib

# SMTP server
dnf install postfix
```

Módulos do Python

Dica: Estamos usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você não estiver familiarizado com ele, consulte [User Guide](#) do virtualenv.

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, você precisa reconstruir o pacote em seu sistema:

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

Configurando o Weblate

Nota: O seguinte assume que o virtualenv utilizado pela Weblate é ativado (executando `. ~/weblate-env/bin/bin/activate`). Caso contrário, especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas você vai querer mais mudanças para configuração de produção, veja [Ajustando a configuração](#).

3. Crie o banco de dados e sua estrutura para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para uma configuração pronta para produção):

```
weblate migrate
```

4. Crie uma conta para o usuário administrador e copie sua senha para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

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

```
weblate collectstatic
```

6. Compacte os arquivos JavaScript e CSS (opcional, veja [Comprimindo os ativos do cliente](#)):

```
weblate compress
```

7. Inicie os workers do Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. [Tarefas de fundo usando Celery](#) tem mais informações:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

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

```
weblate runserver
```

Após a instalação

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

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Faça o login com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando virtualenv do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com Ctrl+C.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` (veja [Interface de gerenciamento](#)) ou usando **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que você precisa especificar aqui é o nome do projeto e seu site.

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

Os campos importantes aqui são: *Nome do componente*, *Repositório do código-fonte* e *Máscara de arquivo* para localizar arquivos traduzíveis. Weblate tem suporte a uma ampla gama de formatos, incluindo *GNU gettext*, *Recurso de textos de Android*, *Textos de Apple iOS*, *Propriedades Java*, *Formato stringsdict* ou *Formato Fluent*, consulte [Formatos de arquivos suportados](#) para mais detalhes.

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

Instalando no macOS

Requisitos de hardware

O Weblate deve funcionar em qualquer hardware contemporâneo sem problemas. A seguir está a configuração mínima necessária para executar o Weblate em um único host (Weblate, banco de dados e servidor web):

- 3 GB of RAM
- 2 núcleos de CPU
- 1 GB de espaço de armazenamento

Quanto mais memória melhor – ele é usada para cache em todos os níveis (sistema de arquivos, banco de dados e Weblate).

Muitos usuários simultâneos aumentam a quantidade de núcleos de CPU necessários. Para centenas de componentes de tradução é recomendado pelo menos 4 GB de RAM.

O uso típico de armazenamento de banco de dados é de cerca de 300 MB por 1 milhão de palavras hospedadas. O espaço de armazenamento necessário para repositórios clonados varia, mas o Weblate tenta manter seu tamanho mínimo fazendo clones rasos.

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

Instalação

Requisitos do sistema

Instale as dependências necessárias para construir os módulos Python (veja *Requisitos de software*):

```
brew install python pango cairo gobject-introspection libffi glib libyaml
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, dependendo dos recursos que você pretende usar (veja *Dependências opcionais*):

```
brew install tesseract
```

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

As instruções de instalação local:

```
# Web server option 1: NGINX and uWSGI
brew install nginx uwsgi

# Web server option 2: Apache with ``mod_wsgi``
brew install httpd

# Caching backend: Redis
brew install redis
```

(continua na próxima página)

(continuação da página anterior)

```
# Database server: PostgreSQL
brew install postgresql
```

Módulos do Python

Dica: Estamos usando o virtualenv para instalar o Weblate em um ambiente separado do seu sistema. Se você não estiver familiarizado com ele, consulte [User Guide](#) do virtualenv.

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, você precisa reconstruir o pacote em seu sistema:

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

Configurando o Weblate

Nota: O seguinte assume que o virtualenv utilizado pela Weblate é ativado (executando `. ~/weblate-env/bin/bin/activate`). Caso contrário, especificar o caminho completo para o comando **weblate** como `~/weblate-env/bin/weblate`.

1. Copie o arquivo `~/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 arquivo `settings.py` de acordo com sua preferência. Você vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas você vai querer mais mudanças para configuração de produção, veja [Ajustando a configuração](#).
3. Crie o banco de dados e sua estrutura para Weblate (as configurações de exemplo usam PostgreSQL, confira [Configuração de banco de dados para o Weblate](#) para uma configuração pronta para produção):

```
weblate migrate
```

4. Crie uma conta para o usuário administrador e copie sua senha para a área de transferência e salve-a para uso posterior:

```
weblate createadmin
```

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

```
weblate collectstatic
```

6. Compacte os arquivos JavaScript e CSS (opcional, veja [Comprimindo os ativos do cliente](#)):

```
weblate compress
```

7. Inicie os workers do Celery. Isso não é necessário para fins de desenvolvimento, mas é altamente recomendado caso contrário. [Tarefas de fundo usando Celery](#) tem mais informações:

```
~/weblate-env/lib/python3.9/site-packages/weblate/examples/celery start
```

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

```
weblate runserver
```

Após a instalação

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

- Agora você pode acessar o Weblate em `http://localhost:8000/`.
- Faça o login com as credenciais de administrador obtidas durante a instalação ou registre-se com novos usuários.
- Agora você pode executar comandos Weblate usando o comando **weblate** quando `virtualenv` do Weblate está ativo, veja [Comandos de gerência](#).
- Você pode parar o servidor de teste com `Ctrl+C`.
- Revise os possíveis problemas com sua instalação na URL `/manage/performance/` (veja [Interface de gerenciamento](#)) ou usando **weblate check --deploy**, veja [Configuração de produção](#).

Adicionando tradução

1. Abra a interface de administração (`http://localhost:8000/create/project/`) e crie o projeto que deseja traduzir. Veja [Configuração de projeto](#) para mais detalhes.

Tudo que você precisa especificar aqui é o nome do projeto e seu site.

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

Os campos importantes aqui são: *Nome do componente*, *Repositório do código-fonte* e *Máscara de arquivo* para localizar arquivos traduzíveis. Weblate tem suporte a uma ampla gama de formatos, incluindo *GNU gettext*, *Recurso de textos de Android*, *Textos de Apple iOS*, *Propriedades Java*, *Formato stringsdict* ou *Formato Fluent*, consulte [Formatos de arquivos suportados](#) para mais detalhes.

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

Instalando a partir do código-fonte

1. Siga as instruções de instalação do seu sistema primeiro até a primeira instalação:

- *Instalando no Debian e Ubuntu*
- *Instalando no SUSE e openSUSE*
- *Instalando no RedHat, Fedora e CentOS*

2. Pegue o código-fonte mais recente do Weblate usando Git (ou baixe um tarball e descompacte-o):

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

Alternativamente, você pode usar arquivos lançados. Você pode baixá-los em nosso site <<https://weblate.org/>>. Esses downloads são assinados criptograficamente, consulte *Verificando assinaturas de lançamento*.

3. Instale o código atual do Weblate no virtualenv:

```
. ~/weblate-env/bin/activate  
pip install -e weblate-src
```

4. Copie `weblate/settings_example.py` para `weblate/settings.py`.
5. Ajuste os valores no novo arquivo `settings.py` de acordo com sua preferência. Você vai precisar fornecer pelo menos as credenciais de banco de dados e a chave secreta Django, mas você vai querer mais mudanças para configuração de produção, veja *Ajustando a configuração*.
6. Crie o banco de dados usado pelo Weblate, veja *Configuração de banco de dados para o Weblate*.
7. Construa tabelas Django, arquivos estáticos e dados iniciais (veja *Preenchendo o banco de dados e Servindo arquivos estáticos*):

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

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

Instalando no OpenShift

Com o modelo OpenShift Weblate, você pode colocar sua instância pessoal do Weblate em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão e declarações de volume persistentes são usadas.

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

Instalação

Os exemplos a seguir presumem que você tem um ambiente OpenShift v3.x funcional, com a ferramenta cliente `oc` instalada. Confira a documentação do OpenShift para obter instruções.

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 você configure um servidor PostgreSQL externo.

Console web

Copie o conteúdo bruto de `template.yml` e importe-o para seu projeto, então use o botão `Create` no console web do OpenShift para criar seu aplicativo. O console web solicitará os valores de todos os parâmetros usados pelo modelo.

CLI

Para enviar o modelo Weblate para a biblioteca de modelos do seu projeto atual, passe o arquivo `template.yml` com o seguinte comando:

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

O modelo agora está disponível para seleção usando o console web ou o CLI.

Parâmetros

Os parâmetros que você pode substituir estão listados na seção de parâmetros do modelo. Você pode listá-los com a CLI usando o seguinte comando e especificando o arquivo a ser usado:

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

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

Provisionamento

Você também pode usar a CLI para processar modelos e usar a configuração gerada para criar objetos imediatamente.

```
$ oc process -f https://raw.githubusercontent.com/WeblateOrg/openshift/main/
↪template.yml \
  -p APPLICATION_NAME=weblate \
  -p WEBLATE_VERSION=4.3.1-1 \
  -p WEBLATE_SITE_DOMAIN=weblate.app-openshift.example.com \
  -p POSTGRESQL_IMAGE=docker-registry.default.svc:5000/openshift/postgresql:9.6 \
  -p REDIS_IMAGE=docker-registry.default.svc:5000/openshift/redis:3.2 \
  | oc create -f
```

A instância do Weblate deve estar disponível após a migração e implantação com sucesso no parâmetro `WEBLATE_SITE_DOMAIN` especificado.

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

Para redefinir a senha do *admin*, reinicie o contêiner com `WEBLATE_ADMIN_PASSWORD` definido para a nova senha no respectivo `Secret`.

Eliminar

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

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

Configuração

Ao processar o modelo, um respectivo ConfigMap será criado e poderá ser usado para personalizar a imagem Weblate. O ConfigMap é montado diretamente como variáveis de ambiente e aciona uma nova implantação toda vez que é alterado. Para mais opções de configuração, consulte *Variáveis de ambiente do Docker* para uma lista completa de variáveis de ambiente.

Instalando no Kubernetes

Nota: Este guia procura colaboradores experientes com o Kubernetes para abordar a configuração com mais detalhes.

Com o chart de Kubernetes Helm, você pode colocar sua instância pessoal do Weblate em funcionamento em segundos. Todas as dependências do Weblate já estão incluídas. PostgreSQL é configurado como o banco de dados padrão e declarações de volume persistentes são usadas.

Você pode encontrar o chart em <https://github.com/WeblateOrg/helm/> e pode ser exibido em <https://artifacthub.io/packages/helm/weblate/weblate>.

Instalação

```
helm repo add weblate https://helm.weblate.org
helm install my-release weblate/weblate
```

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 você:

- *Instalando usando Docker*, recomendado para configurações de produção.
- Instalação com virtualenv, recomendada para configurações de produção:
 - *Instalando no Debian e Ubuntu*
 - *Instalando no SUSE e openSUSE*
 - *Instalando no RedHat, Fedora e CentOS*
 - *Instalando no macOS*

- *Instalando a partir do código-fonte*, recomendado para o desenvolvimento.
- *Instalando no OpenShift*
- *Instalando no Kubernetes*

2.1.2 Requisitos de software

Sistema operacional

Weblate é conhecido por funcionar no Linux, FreeBSD e macOS. Outros sistemas como o Unix provavelmente funcionarão também.

O Weblate não é suportado no Windows. Mas ainda pode funcionar e patches são aceitos alegremente.

Outros serviços

Weblate está usando outros serviços para sua operação. Você precisará pelo menos seguir os serviços em execução:

- Servidor de banco de dados PostgreSQL, consulte *Configuração de banco de dados para o Weblate*.
- Servidor Redis para cache e fila de tarefas, consulte *Tarefas de fundo usando Celery*.
- Servidor SMTP para e-mail de saída, consulte *Configuração de e-mail de saída*.

Dependências Python

Weblate é escrito em [Python](#) e tem suporte a Python 3.6 ou mais novo. Você pode instalar dependências usando pip ou de seus pacotes de distribuição, a lista completa está disponível em `requirements.txt`.

As dependências mais notáveis:

Django

<https://www.djangoproject.com/>

Celery

<https://docs.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 seguintes módulos são necessários para alguns recursos do Weblate. Você pode encontrar todos eles em `requirements-optional.txt`.

Mercurial (opcional para suporte a repositórios *Mercurial*)

<https://www.mercurial-scm.org/>

phply (opcional para *Textos do PHP*)

<https://github.com/viraptor/phply>

tesseract (opcional para OCR em *Contexto visual para textos*)

<https://github.com/sirfz/tesseract>

python-akismet (opcional para *Proteção contra spam*)

<https://github.com/Nekmo/python-akismet>

ruamel.yaml (opcional para *Arquivos YAML*)

<https://pypi.org/project/ruamel.yaml/>

Zeep (opcional para *Terminologia Microsoft*)

<https://docs.python-zeep.org/>

aeidon (opcional para *Arquivos de legenda*)

<https://pypi.org/project/aeidon/>

fluent.syntax (opcional para *Formato Fluent*)

<https://projectfluent.org/>

Dica: Ao instalar usando pip, você pode especificar diretamente os recursos desejados ao instalar:

```
pip install "Weblate[PHP,Fluent]"
```

Ou você pode instalar o Weblate com todos os recursos opcionais:

```
pip install "Weblate[all]"
```

Ou você pode instalar o Weblate sem quaisquer recursos opcionais:

```
pip install Weblate
```

Dependências de backend de banco de dados

O Weblate tem suporte a PostgreSQL, MySQL e MariaDB, consulte *Configuração de banco de dados para o Weblate* e a documentação dos backends para mais detalhes.

Outros requisitos do sistema

As seguintes dependências devem ser instaladas no sistema:

Git

<https://git-scm.com/>

Pango, Cairo e arquivos de cabeçalho relacionados e dados de introspecção GObject

<https://cairographics.org/>, <https://pango.gnome.org/>, veja *Pango e Cairo*

git-review (opcional para suporte a Gerrit)

<https://pypi.org/project/git-review/>

git-svn (opcional para suporte a Subversion)

<https://git-scm.com/docs/git-svn>

tesseract e seus dados (opcional para OCR de capturas de tela)

<https://github.com/tesseract-ocr/tesseract>

licensee (opcional para detectar a licença ao criar o componente)

<https://github.com/licensee/licensee>

Dependências de tempo de compilação

Para compilar alguns das *dependências Python*, você pode precisar instalar suas dependências. Isso depende de como você instalá-las, por isso consulte pacotes individuais para obter documentação. Você não precisará deles se usar `Wheels` pré-compilado durante a instalação usando `pip` ou quando você usar pacotes de distribuição.

Pango e Cairo

Alterado na versão 3.7.

O Weblate usa Pango e Cairo para renderizar widgets bitmap (ver *promotion*) e verificações de renderização (ver *Gerenciando fontes*). Para instalar corretamente as ligações Python para aqueles que você precisa instalar bibliotecas de sistemas primeiro - você precisa tanto do Cairo quanto do Pango, que por sua vez precisam de GLib. Todos esses devem ser instalados com arquivos de desenvolvimento e dados de introspecção GObject.

2.1.3 Verificando assinaturas de lançamento

Os lançamentos do Weblate é criptograficamente assinados pelo desenvolvedor que o lançou. Atualmente este é Michal Čihař. A impressão digital da chave PGP é:

```
63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

e você pode obter mais informações de identificação de <https://keybase.io/nijel>.

Você deve verificar se a assinatura corresponde ao arquivo que você baixou. Desta forma, você pode ter certeza de que está usando o mesmo código que foi lançado. Você também deve verificar a data da assinatura para ter certeza de que você baixou a versão mais recente.

Cada arquivo é acompanhado com arquivos `.asc`, os quais contêm a assinatura PGP para ele. Uma vez que você tenha ambos na mesma pasta, você pode verificar a assinatura:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:          using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Can't check signature: public key not found
```

Como você pode ver, o GPG reclama que não conhece a chave pública. Neste ponto você deve fazer um dos seguintes passos:

- Use `wkd` para baixar a chave:

```
$ gpg --auto-key-locate wkd --locate-keys michal@cihar.com
pub  rsa4096 2009-06-17 [SC]
    63CB1DF1EF12CF2AC0EE5A329C27B31342B7511D
uid          [ultimate] Michal Čihař <michal@cihar.com>
uid          [ultimate] Michal Čihař <nijel@debian.org>
uid          [ultimate] [jpeg image of size 8848]
uid          [ultimate] Michal Čihař (Braiiins) <michal.cihar@braiins.cz>
sub  rsa4096 2009-06-17 [E]
sub  rsa4096 2015-09-09 [S]
```

- Baixe o chaveiro do [servidor do Michal](#) e importe-o com:

```
$ gpg --import wmxth3chu9jfxdxywj1skpmhsj311mzm
```

- Baixe e importe a chave de um dos principais servidores:

```
$ gpg --keyserver hkp://pgp.mit.edu --recv-keys 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: key 9C27B31342B7511D: "Michal Čihař <michal@cihar.com>" imported
gpg: Total number processed: 1
gpg:                          unchanged: 1
```

Isso vai melhorar um pouco a situação - neste momento, você pode verificar que a assinatura da chave dada está correta, mas você ainda não pode confiar no nome usado na chave:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Ne 3. března 2019, 16:43:15 CET
gpg:      using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:      aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:      aka "[jpeg image of size 8848]" [ultimate]
gpg:      aka "Michal Čihař (Brains) <michal.cihar@brains.cz>" [ultimate]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:      There is no indication that the signature belongs to the owner.
Primary key fingerprint: 63CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D
```

O problema aqui é que qualquer um poderia emitir a chave com este nome. Você precisa garantir que a chave é realmente propriedade da pessoa mencionada. O Manual de Privacidade do GNU aborda este tópico no capítulo [Validating other keys on your public keyring](#). O método mais confiável é conhecer o desenvolvedor pessoalmente e trocar impressões digitais importantes, no entanto você também pode confiar na rede de confiança. Dessa forma, você pode confiar na chave transitivamente através de assinaturas de outros, que conheceram o desenvolvedor pessoalmente.

Uma vez que a chave seja confiável, o aviso não ocorrerá:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: assuming signed data in 'Weblate-3.5.tar.xz'
gpg: Signature made Sun Mar 3 16:43:15 2019 CET
gpg:      using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: Good signature from "Michal Čihař <michal@cihar.com>" [ultimate]
gpg:      aka "Michal Čihař <nijel@debian.org>" [ultimate]
gpg:      aka "[jpeg image of size 8848]" [ultimate]
gpg:      aka "Michal Čihař (Brains) <michal.cihar@brains.cz>" [ultimate]
```

Se a assinatura for inválida (o arquivo foi alterado), você obterá um erro claro, independentemente do fato de que a chave é confiável ou não:

```
$ gpg --verify Weblate-3.5.tar.xz.asc
gpg: Signature made Sun Mar 3 16:43:15 2019 CET
gpg:      using RSA key 87E673AF83F6C3A0C344C8C3F4AA229D4D58C245
gpg: BAD signature from "Michal Čihař <michal@cihar.com>" [ultimate]
```

2.1.4 Permissões do sistema de arquivos

O processo Weblate precisa ser capaz de ler e escrever para o diretório onde mantém os dados – `DATA_DIR`. Todos os arquivos dentro deste diretório devem ser de propriedade e graváveis pelo usuário que executa todos os processos do Weblate (geralmente WSGI e Celery, veja [Executando servidor](#) e [Tarefas de fundo usando Celery](#)).

A configuração padrão coloca-os na mesma árvore que os fontes do Weblate, no entanto, você pode preferir movê-los para um local melhor, como `/var/lib/weblate`.

O Weblate tenta criar esses diretórios automaticamente, mas ele falhará quando não tiver permissões para fazê-lo.

Você também deve tomar cuidado ao executar *Comandos de gerência*, pois eles devem ser executados sob o mesmo usuário que o Weblate em si está sendo executado, caso contrário, permissões em alguns arquivos podem estar erradas.

No contêiner Docker, todos os arquivos no volume `/app/data` tem de ter como dono o usuário `weblate` dentro do contêiner (UID 1000).

Ver também:

Servindo arquivos estáticos

2.1.5 Configuração de banco de dados para o Weblate

Recomenda-se executar o Weblate com um servidor de banco de dados PostgreSQL.

Ver também:

Usar um poderoso mecanismo de banco de dados, Banco de Dados, Migrando de outros bancos de dados para o PostgreSQL

PostgreSQL

PostgreSQL é geralmente a melhor escolha para sites baseados em Django. É o banco de dados de referência usado para implementar a camada de banco de dados Django.

Nota: O Weblate usa uma extensão de trigram que deve ser instalada separadamente em alguns casos. Procure por `postgresql-contrib` ou um pacote com nome similar.

Ver também:

PostgreSQL notas

Criando um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate em um banco de dados separado e separar a conta do usuário:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser --superuser --pwprompt weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Dica: Se você não quiser fazer do usuário do Weblate um superusuário no PostgreSQL, você pode omitir isso. Nesse caso, você terá que executar algumas das etapas de migração manualmente como um superusuário PostgreSQL no esquema Weblate usará:

```
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH SCHEMA weblate;
CREATE EXTENSION IF NOT EXISTS btree_gin WITH SCHEMA weblate;
```

Configurando Weblate para usar PostgreSQL

O trecho de `settings.py` para PostgreSQL:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.postgresql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Name of role to alter to set parameters in PostgreSQL,
        # use in case role name is different than user used for authentication.
        # "ALTER_ROLE": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "database.example.com",
        # Set to empty string for default
        "PORT": "",
    }
}
```

A migração do banco de dados executa `ALTER ROLE` na função de banco de dados usada pelo Weblate. Na maioria dos casos, o nome da função corresponde ao nome de usuário. Em configurações mais complexas, o nome da função é diferente do nome de usuário e você obterá um erro sobre a função não existente durante a migração do banco de dados (`psycopg2.errors.UndefinedObject: role "weblate@hostname" does not exist`). Isso é conhecido por acontecer com o Azure Database para PostgreSQL, mas não está limitado a este ambiente. Defina `ALTER_ROLE` para alterar o nome da função que o Weblate deve alterar durante a migração do banco de dados.

MySQL e MariaDB

Dica: Alguns recursos do Weblate terão melhor desempenho com *PostgreSQL*. Isso inclui a memória de pesquisa e tradução, que ambos utilizam recursos de texto completo no banco de dados e a implementação do PostgreSQL é superior.

Weblate também pode ser usado com MySQL ou MariaDB, consulte [MySQL notes](#) e [MariaDB notes](#) para ressalvas ao uso do Django com elas. or causa das limitações, é recomendado usar *PostgreSQL* para novas instalações.

O Weblate requer MySQL pelo menos 5.7.8 ou MariaDB pelo menos 10.2.7.

A seguinte configuração é recomendada para Weblate:

- Use o conjunto de caracteres `utf8mb4` para permitir a representação de planos Unicode mais altos (por exemplo, emojis).
- Configure o servidor com `innodb_large_prefix` para permitir índices mais longos em campos de texto.
- Defina o nível de isolamento para `READ COMMITTED`.
- O modo SQL deve ser definido como `STRICT_TRANS_TABLES`.

MySQL 8.x, MariaDB 10.5.x 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 você esteja planejando ter um grande número de usuários simultâneos acessando o sistema. Consulte a documentação de vários fornecedores sobre esses detalhes.

É absolutamente crítico para reduzir problemas durante a instalação que a configuração `innodb_file_per_table` esteja configurada corretamente e o MySQL/MariaDB reiniciado antes de você iniciar a instalação do Weblate.

```
[mysqld]
character-set-server = utf8mb4
character-set-client = utf8mb4
collation-server = utf8mb4_unicode_ci

datadir=/var/lib/mysql

log-error=/var/log/mariadb/mariadb.log

innodb_large_prefix=1
innodb_file_format=Barracuda
innodb_file_per_table=1
innodb_buffer_pool_size=2G
sql_mode=STRICT_TRANS_TABLES
```

Dica: No caso de você estar recebendo o erro #1071 - Specified key was too long; max key length is 767 bytes, atualize sua configuração para incluir as configurações `innodb` acima e reinicie sua instalação.

Dica: No caso de você estar recebendo o erro #2006 - MySQL server has gone away, configurar `CONN_MAX_AGE` pode ajudar.

Configurando Weblate para usar MySQL/MariaDB

O trecho de `settings.py` para MySQL e MariaDB:

```
DATABASES = {
    "default": {
        # Database engine
        "ENGINE": "django.db.backends.mysql",
        # Database name
        "NAME": "weblate",
        # Database user
        "USER": "weblate",
        # Database password
        "PASSWORD": "password",
        # Set to empty string for localhost
        "HOST": "127.0.0.1",
        # Set to empty string for default
        "PORT": "3306",
        # In case you wish to use additional
        # connection options
        "OPTIONS": {},
    }
}
```

Você também deve criar a conta de usuário `weblate` no MySQL ou MariaDB antes de iniciar a instalação. Use os comandos abaixo para fazer isso:

```
GRANT ALL ON weblate.* to 'weblate'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

2.1.6 Outras configurações

Configuração de e-mail de saída

O Weblate envia e-mails em várias ocasiões - para ativação de conta e sobre várias notificações configuradas pelos usuários. Para isso, ele precisa de acesso a um servidor SMTP.

A configuração do servidor de e-mail é configurada usando essas configurações: `EMAIL_HOST`, `EMAIL_HOST_PASSWORD`, `EMAIL_USE_SSL`, `EMAIL_USE_TLS`, `EMAIL_HOST_USER` e `EMAIL_PORT`. Seus nomes são bastante autoexplicativos, mas você pode encontrar mais informações na documentação do Django.

Dica: Caso você tenha erro sobre autenticação não suportada (por exemplo, `SMTP AUTH extension not supported by server`), é provavelmente causada pelo uso de conexão insegura e o servidor se recusa a autenticar dessa forma. Tente ativar `EMAIL_USE_TLS` nesse caso.

Ver também:

Não recebendo e-mails do Weblate, Configurando e-mail de saída em contêiner Docker

Executando por trás de um proxy reverso

Vários recursos no Weblate dependem de ser capaz de obter endereço IP do cliente. Isso inclui *Limitação de taxa*, *Proteção contra spam* ou *Registro de auditoria*.

Na configuração padrão, o Weblate analisa o endereço IP de `REMOTE_ADDR` que é definido pelo manipulador WSGI.

No caso de você estiver usando um proxy reverso, este campo provavelmente conterá seu endereço. Você precisa configurar o Weblate para confiar em cabeçalhos HTTP adicionais e analisar o endereço IP destes. Isso não pode ser ativado por padrão, pois permitiria falsificação de endereço IP para instalações que não usam um proxy reverso. Habilitar `IP_BEHIND_REVERSE_PROXY` pode ser suficiente para as configurações mais usuais, mas você pode precisar ajustar `IP_PROXY_HEADER` e `IP_PROXY_OFFSET` também.

Outra coisa a ajustar é o cabeçalho `Host`. Deve corresponder ao que está configurado como `SITE_DOMAIN`. Configuração adicional pode ser necessária em seu proxy reverso (por exemplo, use `ProxyPreserveHost On` para Apache ou `proxy_set_header Host $host;` com nginx).

Ver também:

Proteção contra spam, *Limitação de taxa*, *Registro de auditoria*, `IP_BEHIND_REVERSE_PROXY`, `IP_PROXY_HEADER`, `IP_PROXY_OFFSET`, `SECURE_PROXY_SSL_HEADER`

Proxy HTTP

O Weblate executa comandos VCS e aqueles que aceitam a configuração proxy do ambiente. A abordagem recomendada é definir configurações de proxy em `settings.py`:

```
import os

os.environ["http_proxy"] = "http://proxy.example.com:8080"
os.environ["HTTPS_PROXY"] = "http://proxy.example.com:8080"
```

Ver também:

Variáveis de ambiente de proxy

2.1.7 Ajustando a configuração

Ver também:

Amostra de configuração

Copie `weblate/settings_example.py` para `weblate/settings.py` e ajuste-o para corresponder à configuração. Você provavelmente vai querer ajustar as seguintes opções: `ADMINS`

Lista de administradores de sites para receber notificações quando algo dá errado, por exemplo, notificações em mesclagens fracassadas ou erros de Django.

Ver também:

`ADMINS`, *Configurar corretamente administradores*

`ALLOWED_HOSTS`

Você precisa definir isso para listar os hosts que seu site deve servir. Por exemplo:

```
ALLOWED_HOSTS = ["demo.weblate.org"]
```

Alternativamente, você pode incluir curinga:

```
ALLOWED_HOSTS = ["*"]
```

Ver também:

`ALLOWED_HOSTS`, `WEBLATE_ALLOWED_HOSTS`, *Configuração de hosts permitidos*

`SESSION_ENGINE`

Configure como suas sessões serão armazenadas. Caso você mantenha o mecanismo de backend do banco de dados padrão, você deve agendar: **`weblate clearsessions`** para remover dados de sessão obsoletos do banco de dados.

Se você estiver usando o Redis como cache (veja *Habilitar o cache*) é recomendado usá-lo para sessões também:

```
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
```

Ver também:

Configuring the session engine, `SESSION_ENGINE`

`DATABASES`

Conectividade ao servidor de banco de dados, verifique a documentação do Django para obter mais detalhes.

Ver também:

Configuração de banco de dados para o Weblate, `DATABASES`, *Banco de Dados*

`DEBUG`

Desabilite isso para qualquer servidor de produção. Com o modo depuração ativado, o Django mostrará backtraces em caso de erro aos usuários, quando você desabilitá-lo, erros serão enviados por e-mail para `ADMINS` (veja acima).

O modo depuração também diminui o Weblate, já que o Django armazena muito mais informações internamente neste caso.

Ver também:

`DEBUG`, *Desabilitar o modo de depuração*

`DEFAULT_FROM_EMAIL`

Endereço de remetente de e-mail para e-mail de saída, por exemplo, e-mails de registro.

Ver também:

`DEFAULT_FROM_EMAIL`

`SECRET_KEY`

Chave usada por Django para assinar algumas informações em cookies, consulte [Chave secreta do Django](#) para obter mais informações.

Ver também:

`SECRET_KEY`

`SERVER_EMAIL`

E-mail usado como endereço de remetente para envio de e-mails ao administrador, por exemplo, notificações em mensagens fracassadas.

Ver também:

`SERVER_EMAIL`

2.1.8 Preenchendo o banco de dados

Depois que sua configuração estiver pronta, você pode executar `weblate migrate` para criar a estrutura do banco de dados. Agora você deve ser capaz de criar projetos de tradução usando a interface administrativa.

Caso você queira executar uma instalação não interativamente, você pode usar `weblate migrate --noinput` e, em seguida, criar um usuário administrativo usando `createadmin` comando.

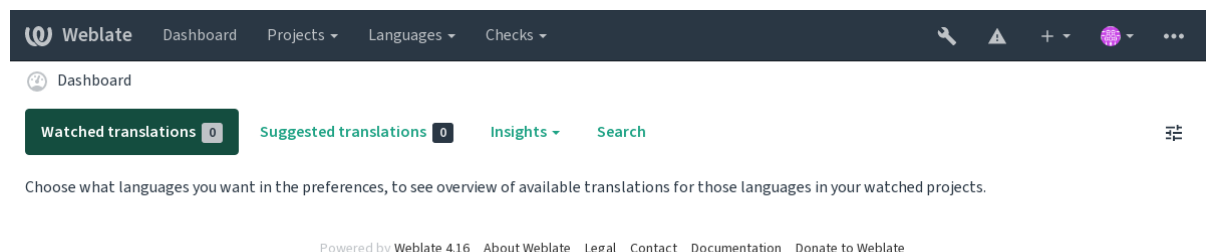
Uma vez feito, você também deve verificar o *Relatório de desempenho* na interface administrativa, o que lhe dará dicas de configuração potencial não ideal em seu site.

Ver também:

[Configuração](#), [Lista de privilégios e funções embutidas](#)

2.1.9 Configuração de produção

Para uma configuração de produção, você deve realizar ajustes descritos nas seções a seguir. As configurações mais críticas acionarão um aviso, que é indicado por um ponto de exclamação na barra superior se conectado como um superusuário:



Também é recomendado inspecionar verificações desencadeadas por Django (embora você possa não precisar corrigir todas elas):

```
weblate check --deploy
```

Você também pode revisar a mesma lista de verificação na [Interface de gerenciamento](#).

Ver também:

[Lista de verificação para distribuição](#)

Desabilitar o modo de depuração

Desabilite o modo de depuração do Django (*DEBUG*) com:

```
DEBUG = False
```

Com o modo depuração ativado, o Django armazena todas as consultas executadas e mostra aos usuários atrasos de erros, o que não é desejado em uma configuração de produção.

Ver também:

Ajustando a configuração

Configurar corretamente administradores

Defina os endereços de administração corretos para a configuração *ADMINS* para definir quem receberá e-mails caso algo dê errado no servidor, por exemplo:

```
ADMINS = (("Your Name", "your_email@example.com"),)
```

Ver também:

Ajustando a configuração

Definir domínio correto do site

Ajuste o nome e o domínio do site na interface administrativa, caso contrário, links no RSS ou e-mails de registro não funcionarão. Isso é configurado usando *SITE_DOMAIN* que deve conter o nome de domínio do site.

Alterado na versão 4.2: Antes da versão 4.2, a estrutura de sites do Django era usada em vez disso, consulte [The “sites” framework](#).

Ver também:

Configuração de hosts permitidos, Configurar corretamente HTTPS *SITE_DOMAIN*, *WEBLATE_SITE_DOMAIN*, *ENABLE_HTTPS*

Configurar corretamente HTTPS

É fortemente recomendado executar Weblate usando o protocolo HTTPS criptografado. Depois de habilitá-lo, você deve definir *ENABLE_HTTPS* nas configurações:

```
ENABLE_HTTPS = True
```

Dica: Você pode querer configurar o HSTS também, consulte [SSL/HTTPS](#) para obter mais detalhes.

Ver também:

ENABLE_HTTPS, Configuração de hosts permitidos, Definir domínio correto do site

Definir corretamente `SECURE_HSTS_SECONDS`

Se o seu site for servido sobre SSL, você deve considerar definir um valor para `SECURE_HSTS_SECONDS` no `settings.py` para habilitar HTTP Strict Transport Security. Por padrão, ele está definido para 0 como mostrado abaixo.

```
SECURE_HSTS_SECONDS = 0
```

Se definido como um valor inteiro não-zero, o cabeçalho `django.middleware.security.SecurityMiddleware` define o cabeçalho HTTP Strict Transport Security em todas as respostas que ainda não o possuem.

Aviso: Definir isso incorretamente pode quebrar irreversivelmente (por algum tempo) seu site. Leia primeiro a documentação [HTTP Strict Transport Security](#).

Usar um poderoso mecanismo de banco de dados

- Por favor, use PostgreSQL para um ambiente de produção, consulte *Configuração de banco de dados para o Weblate* para obter mais informações.
- Use um local adjacente para executar o servidor de banco de dados, caso contrário, o desempenho ou confiabilidade da rede podem arruinar sua experiência com o Weblate.
- Verifique o desempenho do servidor de banco de dados ou ajuste sua configuração, por exemplo, usando [PG-Tune](#).

Ver também:

Configuração de banco de dados para o Weblate, Migrando de outros bancos de dados para o PostgreSQL, Ajustando a configuração, Banco de Dados

Habilitar o cache

Se possível, use Redis do Django ajustando a variável de configuração `CACHES`, por exemplo:

```
CACHES = {
    "default": {
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "redis://127.0.0.1:6379/0",
        # If redis is running on same host as Weblate, you might
        # want to use unix sockets instead:
        # 'LOCATION': 'unix:///var/run/redis/redis.sock?db=0',
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
}
```

Dica: Caso você altere as configurações do Redis para o cache, pode ser necessário ajustá-las também para o Celery, consulte *Tarefas de fundo usando Celery*.

Ver também:

Cache de avatares, O framework de “cache” do Django

Cache de avatares

Além do cache de Django, Weblate realiza cache de avatares. Recomenda-se usar um cache separado, baseado em arquivos para este fim:

```
CACHES = {
    "default": {
        # Default caching backend setup, see above
        "BACKEND": "django_redis.cache.RedisCache",
        "LOCATION": "unix:///var/run/redis/redis.sock?db=0",
        "OPTIONS": {
            "CLIENT_CLASS": "django_redis.client.DefaultClient",
            "PARSER_CLASS": "redis.connection.HiredisParser",
        },
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 604800,
        "OPTIONS": {
            "MAX_ENTRIES": 1000,
        },
    },
}
```

Ver também:

[`ENABLE_AVATARS`](#), [`AVATAR_URL_PREFIX`](#), [Avatares](#), [Habilitar o cache](#), [O framework de “cache” do Django](#)

Configurar envio de e-mail

O Weblate precisa enviar e-mails em várias ocasiões, e esses e-mails devem ter um endereço de remetente correto, por favor, configure `SERVER_EMAIL` e `DEFAULT_FROM_EMAIL` para combinar com o seu ambiente, por exemplo:

```
SERVER_EMAIL = "admin@example.org"
DEFAULT_FROM_EMAIL = "weblate@example.org"
```

Nota: Para desabilitar o envio de e-mails pelo Weblate, defina `EMAIL_BACKEND` para `django.core.mail.backends.dummy.EmailBackend`.

Isso desabilitará *toda* entrega de e-mail, incluindo e-mails de registro ou redefinição de senha.

Ver também:

[Ajustando a configuração](#), [Configuração de e-mail de saída](#), [`EMAIL_BACKEND`](#), [`DEFAULT_FROM_EMAIL`](#), [`SERVER_EMAIL`](#)

Configuração de hosts permitidos

Django requer `ALLOWED_HOSTS` para manter uma lista de nomes de domínio que seu site pode servir, deixando-o vazio bloqueará quaisquer solicitações.

Caso isso não esteja configurado para corresponder ao seu servidor HTTP, você terá erros como Invalid HTTP_HOST header: '1.1.1.1'. You may need to add '1.1.1.1' to ALLOWED_HOSTS.

Dica: No contêiner Docker, isso está disponível como `WEBLATE_ALLOWED_HOSTS`.

Ver também:

[`ALLOWED_HOSTS`](#), [`WEBLATE_ALLOWED_HOSTS`](#), [Definir domínio correto do site](#)

Chave secreta do Django

A configuração `SECRET_KEY` é usada pelo Django para assinar cookies, e você deve realmente gerar seu próprio valor em vez de usar o da configuração do exemplo.

Você pode gerar uma nova chave usando `weblate/examples/generate-secret-key`, que vem com o Weblate.

Ver também:

[`SECRET_KEY`](#)

Executando tarefas de manutenção

Para um desempenho ideal, é uma boa ideia executar algumas tarefas de manutenção em segundo plano. Isso agora é feito automaticamente por [Tarefas de fundo usando Celery](#) e cobre as seguintes tarefas:

- Verificação de saúde da configuração (de hora em hora).
- Realização de commits de alterações pendentes (de hora em hora), consulte [Commits adiados](#) e `commit_pending`.
- Atualização de alertas de componentes (diariamente).
- Atualização dos ramos remotos (*nightly*), consulte [AUTO_UPDATE](#).
- Backup de memória de tradução para JSON (diariamente), consulte `dump_memory`.
- Tarefas de manutenção de texto completo e banco de dados (tarefas diárias e semanais), consulte `cleanup-trans`.

Alterado na versão 3.2: Desde a versão 3.2, a maneira padrão de executar essas tarefas é usar o Celery e o Weblate já vem com a configuração adequada, consulte [Tarefas de fundo usando Celery](#).

Codificação e localidades do sistema

As localidades do sistema devem ser configuradas para UTF-8. Na maioria das distribuições Linux, esta é a configuração padrão. Caso não seja o caso em seu sistema, altere as localidades para a variante UTF-8.

Por exemplo, editando `/etc/default/locale` e definindo lá `LANG="C.UTF-8"`.

Em alguns casos, os serviços individuais têm configuração separada para locais. 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'
```

Usando autoridade certificadora personalizada

O Weblate verifica os certificados SSL durante as solicitações HTTP. Caso você esteja usando uma autoridade de certificação personalizada que não seja confiável em pacotes padrão, você terá que adicionar seu certificado como confiável.

A abordagem preferida é fazer isso no nível do sistema. Consulte a documentação da sua distro para mais detalhes (por exemplo, no Debian isso pode ser feito colocando o certificado de AC em `/usr/local/share/ca-certificates/` e executando `update-ca-certificates`).

Uma vez feito isso, as ferramentas do sistema confiarão no certificado e isso inclui o Git.

Para código Python, você precisará configurar solicitações para usar o pacote de AC do sistema em vez do enviado com ele. Isso pode ser conseguido colocando seguintes trechos para `settings.py` (o caminho é específico do Debian):

```
import os

os.environ["REQUESTS_CA_BUNDLE"] = "/etc/ssl/certs/ca-certificates.crt"
```

Comprimindo os ativos do cliente

Weblate vem com um monte de arquivos JavaScript e CSS. Por razões de desempenho, é bom comprimi-los antes de enviar para um cliente. Na configuração padrão isso é feito na mosca ao custo de pouca sobrecarga. Em grandes instalações, recomenda-se ativar o modo de compressão offline. Isso precisa ser feito na configuração e a compressão tem que ser acionada em cada atualização do Weblate.

A mudança da configuração é simples ao habilitar `django.conf.settings.COMPRESS_OFFLINE` e configuração `django.conf.settings.COMPRESS_OFFLINE_CONTEXT` (este último já está incluído na configuração do exemplo):

```
COMPRESS_OFFLINE = True
```

Em cada implantação você precisa compactar os arquivos para corresponder à versão atual:

```
weblate compress
```

Dica: A imagem oficial do Docker já tem este recurso habilitado.

Ver também:

[Common Deployment Scenarios](#), [Servindo arquivos estáticos](#)

2.1.10 Executando servidor

Dica: No caso de você não ter experiência com os serviços descritos abaixo, você pode tentar seguir [Instalando usando Docker](#).

Você precisará de vários serviços para executar o Weblate, a configuração recomendada consiste em:

- Servidor de banco de dados (consulte [Configuração de banco de dados para o Weblate](#))
- Servidor de cache (consulte [Habilitar o cache](#))
- Servidor web frontend para arquivos estáticos e terminação SSL (consulte [Servindo arquivos estáticos](#))
- Servidor WSGI para conteúdo dinâmico (consulte [Amostra de configuração para NGINX e uWSGI](#))
- Celery para executar tarefas em segundo plano (consulte [Tarefas de fundo usando Celery](#))

Nota: Existem algumas dependências entre os serviços, por exemplo, o cache e o banco de dados devem estar em execução ao iniciar os processos de Celery ou uwsgi.

Na maioria dos casos, você executará todos os serviços em um único servidor (virtual), mas no caso de sua instalação estar muito carregada, você pode dividir os serviços. A única limitação disso é que os servidores Celery e Wsgi precisam acessar `DATA_DIR`.

Nota: O processo de WSGI deve ser executado sob o mesmo usuário que o processo do Celery, caso contrário, os arquivos em `DATA_DIR` serão armazenados com propriedade mista, levando a problemas de tempo de execução.

Veja também *Permissões do sistema de arquivos* e *Tarefas de fundo usando Celery*.

Executando servidor web

Executar o Weblate não é diferente de executar qualquer outro programa baseado em Django. Django é geralmente executado como uWSGI ou fcgi (consulte exemplos para diferentes servidores web abaixo).

Para fins de teste, você pode usar o servidor web embutido no Django:

```
weblate runserver
```

Aviso: NÃO USE ESTE SERVIDOR EM UMA CONFIGURAÇÃO DE PRODUÇÃO. Ele não passou por auditorias de segurança ou testes de desempenho. Veja também a documentação de Django no `runserver`.

Dica: O servidor embutido do Django serve apenas arquivos estáticos com `DEBUG` ativado, pois é destinado apenas ao desenvolvimento. Para uso da produção, consulte as configurações de wsgi em *Amostra de configuração para NGINX e uWSGI*, *Amostra de configuração para o Apache*, *Amostra de configuração para Apache e Unicorn* e *Servindo arquivos estáticos*.

Servindo arquivos estáticos

Alterado na versão 2.4: Antes da versão 2.4, o Weblate não usava corretamente a estrutura de arquivos estáticos do Django e a configuração era mais complexa.

Django precisa coletar seus arquivos estáticos em um único diretório. Para isso, execute `weblate collectstatic --noinput`. Isso copiará os arquivos estáticos em um diretório especificado pela configuração `STATIC_ROOT` (isso é padrão para um diretório `static` dentro de `DATA_DIR`).

Recomenda-se servir arquivos estáticos diretamente do seu servidor web. Você deve usá-los para os seguintes caminhos:

/static/

Serve arquivos estáticos para Weblate e a interface de administração (definida por `STATIC_ROOT`).

/media/

Usado para envio de mídia pelo usuário (por exemplo, capturas de tela).

/favicon.ico

Deve ser reescrito para reescrever uma regra para servir `/static/favicon.ico`.

Ver também:

Amostra de configuração para NGINX e uWSGI, *Amostra de configuração para o Apache*, *Amostra de configuração para Apache e Unicorn*, *Comprimindo os ativos do cliente*, *How to deploy Django*, *How to deploy static files*

Política de segurança de conteúdo

A configuração padrão do Weblate habilita o middleware `weblate.middleware.SecurityMiddleware` que define cabeçalhos HTTP relacionados à segurança como `Content-Security-Policy` ou `X-XSS-Protection`. Eles são configurados por padrão para funcionar com o Weblate e sua configuração, mas isso pode precisar de personalização para o seu ambiente.

Ver também:

`CSP_SCRIPT_SRC`, `CSP_IMG_SRC`, `CSP_CONNECT_SRC`, `CSP_STYLE_SRC`, `CSP_FONT_SRC`

Amostra de configuração para NGINX e uWSGI

Para executar o servidor web de produção, use o wrapper `wsgi` instalado com Weblate (no caso de ambiente virtual é instalado como `~/weblate-env/lib/python3.9/site-packages/weblate/wsgi.py`). Não se esqueça de definir o caminho de pesquisa Python para seu `virtualenv` também (por exemplo, usando `virtualenv = /home/user/weblate-env` no `uWSGI`).

A configuração a seguir executa o Weblate como `uWSGI` sob o servidor web `NGINX`.

Configuração para `NGINX` (também disponível como `weblate/examples/weblate.nginx.conf`):

```
#
# 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:
    }
}
```

(continua na próxima página)

(continuação da página anterior)

```
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:
#
# - 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
```

(continua na próxima página)

(continuação da página anterior)

```
# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

Ver também:

Como usar Django com uWSGI

Amostra de configuração para o Apache

Recomenda-se o uso de prefork de MPM ao usar WSGI com Weblate.

A configuração a seguir executa o Weblate como WSGI. Você precisa ter habilitado `mod_wsgi` (disponível como `weblate/examples/apache.conf`):

```
#
# VirtualHost for Weblate
#
# 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
```

(continua na próxima página)

(continuação da página anterior)

```

    </Files>
  </Directory>

</VirtualHost>

```

Nota: Weblate precisa do Python 3, então, por favor se certifique que você está executando a variante do Python 3 do modwsgi. Usualmente, está disponível como um pacote separado, por exemplo `libapache2-mod-wsgi-py3`.

Ver também:

Codificação e localidades do sistema, Como usar o Django com Apache e mod_wsgi

Amostra de configuração para Apache e Gunicorn

A configuração a seguir executa o Weblate em Gunicorn e Apache 2.4 (disponível como `weblate/examples/apache.gunicorn.conf`):

```

#
# VirtualHost for Weblate using gunicorn on localhost:8000
#
# 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/

```

(continua na próxima página)

(continuação da página anterior)

```
ProxyPreserveHost On
</VirtualHost>
```

Ver também:

django:tutorial/deployment/wsgi/gunicorn

Rodando Weblate sob o caminho

Novo na versão 1.3.

Recomenda-se o uso de prefork de MPM ao usar WSGI com Weblate.

Uma configuração do Apache de amostra para servir o Weblate sob /weblate. Novamente, usando mod_wsgi (também disponível como weblate/examples/apache-path.conf):

```
#
# VirtualHost for Weblate, running under /weblate path
#
# 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>
```

(continua na próxima página)

(continuação da página anterior)

```
</VirtualHost>
```

Adicionalmente, você irá ter de ajustar o `weblate/settings.py`:

```
URL_PREFIX = "/weblate"
```

2.1.11 Tarefas de fundo usando Celery

Novo na versão 3.2.

Weblate usa Celery para executar tarefas regulares e em segundo plano. Você deve executar um 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 *Ganchos de notificação*).
- Executar tarefas de manutenção regulares como backups, limpezas, extensões diárias ou atualizações (veja *Fazendo backup e movendo 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 usando redis como backend se parece com isso:

```
CELERY_TASK_ALWAYS_EAGER = False
CELERY_BROKER_URL = "redis://localhost:6379"
CELERY_RESULT_BACKEND = CELERY_BROKER_URL
```

Ver também:

Configuração do broker Redis no Celery

Você também deve iniciar o worker do Celery para processar as tarefas e iniciar tarefas programadas, isso pode ser feito diretamente na linha de comando (o que é mais útil ao depurar ou desenvolver):

```
./weblate/examples/celery start
./weblate/examples/celery stop
```

Nota: O processo de Celery deve ser executado sob o mesmo usuário que o Weblate e o processo do WSGI, caso contrário, os arquivos em *DATA_DIR* serão armazenados com propriedade mista, levando a problemas de tempo de execução.

Veja também *Permissões do sistema de arquivos* e *Executando servidor*.

Execução de tarefas de Celery no wsgi usando o modo ansioso

Nota: Isso terá um impacto severo no desempenho da interface da web e interromperá os recursos dependendo do acionamento normal (por exemplo, fazer commit de alterações pendentes, notificações de resumo ou backups).

Para o desenvolvimento, você pode querer 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 você vai querer executar o Celery como um daemon e que é coberto por [Daemonization](#). Para as configurações Linux mais comuns usando systemd, você pode usar os arquivos de exemplo disponibilizados na pasta `examples` listado abaixo.

Unidade do systemd a ser colocada como `/etc/systemd/system/celery-weblate.service`:

```
[Unit]
Description=Celery Service (Weblate)
After=network.target

[Service]
Type=forking
User=weblate
Group=weblate
EnvironmentFile=/etc/default/celery-weblate
WorkingDirectory=/home/weblate
RuntimeDirectory=celery
RuntimeDirectoryPreserve=restart
LogsDirectory=celery
ExecStart=/bin/sh -c '${CELERY_BIN} multi start ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'
ExecStop=/bin/sh -c '${CELERY_BIN} multi stopwait ${CELERYD_NODES} \
  --pidfile=${CELERYD_PID_FILE}'
ExecReload=/bin/sh -c '${CELERY_BIN} multi restart ${CELERYD_NODES} \
  -A ${CELERY_APP} --pidfile=${CELERYD_PID_FILE} \
  --logfile=${CELERYD_LOG_FILE} --loglevel=${CELERYD_LOG_LEVEL} ${CELERYD_OPTS}'

[Install]
WantedBy=multi-user.target
```

Configuração do ambiente a ser colocada como `/etc/default/celery-weblate`:

```
# Name of nodes to start
CELERYD_NODES="celery notify memory backup translate"

# Absolute or relative path to the 'celery' command:
CELERY_BIN="/home/weblate/weblate-env/bin/celery"

# App instance to use
# comment out this line if you don't use an app
CELERY_APP="weblate.utils"

# Extra command-line arguments to the worker,
# increase concurrency if you get weblate.E019
CELERYD_OPTS="--beat:celery --queues:celery=celery --prefetch-multiplier:celery=4 \
```

(continua na próxima página)

(continuação da página anterior)

```
--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 usando **logrotate** a ser colocada como `/etc/logrotate.d/celery`:

```
/var/log/celery/*.log {  
    weekly  
    missingok  
    rotate 12  
    compress  
    notifempty  
}
```

Tarefas periódicas usando a batida do Celery

O Weblate vem com configuração embutida para tarefas programadas. No entanto, você pode definir tarefas adicionais em `settings.py`. Por exemplo, veja [Commits adiados](#).

As tarefas deveriam ser executadas por daemons de batidas do Celery. Caso não esteja funcionando corretamente, pode não estar funcionando ou seu banco de dados foi corrompido. Verifique os registros de inicialização do Celery nesse caso para descobrir a causa raiz.

Monitorando status do Celery

Você pode encontrar o comprimento atual das filas de tarefas do Celery na [Interface de gerenciamento](#) ou você pode usar `celery_queues` na linha de comando. Caso a fila fique muito longa, você também terá erro de configuração na interface administrativa.

Aviso: Os erros do Celery são por padrão apenas conectados ao log do Celery e não são visíveis ao usuário. Caso você queira ter uma visão geral sobre tais falhas, recomenda-se ajustar a configuração para ir [Coletando relatórios de erros](#).

Ver também:

[Monitorando o Weblate](#), [Como posso verificar se o meu Weblate está configurado 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 contêiner Docker tem verificação de saúde embutida usando esta URL.

Para monitorar as métricas do Weblate, você pode usar o ponto final `GET /api/metrics/` da API.

Ver também:

Como posso verificar se o meu Weblate está configurado corretamente?, *Monitorando status do Celery*, *Plugin do Weblate para Munin*

2.1.13 Coletando relatórios de erros

Weblate, como qualquer outro software, pode falhar. Para coletar estados de falha úteis, recomendamos usar serviços de terceiros para coletar tais informações. Isso é especialmente útil no caso de falhas nas tarefas do Celery, que de outra forma só relatariam erro nos logs e você não será notificado sobre eles. O Weblate tem suporte para os seguintes serviços:

Sentry

O Weblate possui suporte embutido para [Sentry](#). Para usá-lo, é suficiente definir `SENTRY_DSN` no `settings.py`:

```
SENTRY_DSN = "https://id@your.sentry.example.com/"
```

Rollbar

O Weblate tem suporte embutido para [Rollbar](#). Para usá-lo, basta seguir instruções para o [notificador de Rollbar para Python](#).

Em suma, você precisa ajustar `settings.py`:

```
# Add rollbar as last middleware:
MIDDLEWARE = [
    # ... other middleware classes ...
    "rollbar.contrib.django.middleware.RollbarNotifierMiddleware",
]

# Configure client access
ROLLBAR = {
    "access_token": "POST_SERVER_ITEM_ACCESS_TOKEN",
    "client_token": "POST_CLIENT_ITEM_ACCESS_TOKEN",
    "environment": "development" if DEBUG else "production",
    "branch": "main",
    "root": "/absolute/path/to/code/root",
}
```

Todo o resto é integrado automaticamente, agora você coletará erros do lado do servidor e do cliente.

2.1.14 Migrando Weblate para outro servidor

Migrar o Weblate para outro servidor deve ser muito fácil, porém armazena dados em poucos locais que você deve migrar cuidadosamente. A melhor abordagem é parar o Weblate para a migração.

Migrando banco de dados

Dependendo do back-end do seu banco de dados, você pode ter várias opções para migrar o banco de dados. A abordagem mais direta é usar ferramentas nativas de banco de dados, pois elas são geralmente as mais eficazes (por exemplo, `mysqldump` ou `pg_dump`). Como alternativa, você pode usar a replicação caso seu banco de dados suporte isso.

Ver também:

Migração entre bancos de dados descritos em *Migrando de outros bancos de dados para o PostgreSQL*.

Migrando repositórios VCS

Os repositórios VCS armazenados em `DATA_DIR` também precisam ser migrados. Você pode simplesmente copiá-los ou usar `rsync` para fazer a migração de forma mais eficaz.

Outras notas

Não se esqueça de mover outros serviços que o Weblate pode ter usado como Redis, tarefas do cron ou backends de autenticação personalizada.

2.2 Implantações de Weblate

O Weblate pode ser facilmente instalado em sua nuvem. Encontre um guia detalhado para sua plataforma:

- *Instalando usando Docker*
- *Instalando no OpenShift*
- *Instalando no Kubernetes*

2.2.1 Implantações de terceiros para Weblate

Nota: As seguintes implantações não são desenvolvidas ou suportadas pela equipe Weblate. Partes da configuração podem variar do que está descrito nesta documentação.

Pilha Weblate para Bitnami

Bitnami provides a Weblate stack for many platforms at <<https://bitnami.com/stack/weblate>>.

Ver também:

Weblate packaged by Bitnami

Pacote de Weblate para Cloudron

Cloudron é uma plataforma para auto-hospedagem de aplicativos da web. Weblate instalado com Cloudron será automaticamente atualizado. O pacote é mantido pela equipe Cloudron em seu [repositório de pacotes Weblate](#).



Weblate no YunoHost

O projeto de auto-hospedagem YunoHost fornece um pacote para Weblate. Uma vez que você tenha a sua instalação YunoHost, você pode instalar o Weblate como qualquer outro aplicativo. Ele fornecerá uma pilha de trabalho completo com backup e restauração, mas você ainda pode ter que editar seu arquivo de configurações para usos específicos.

Você pode usar sua interface de administração ou este botão (ele vai levá-lo ao seu servidor):



Também é possível usar a interface da linha de comando:

```
yunohost app install https://github.com/YunoHost-Apps/weblate_ynh
```

2.3 Atualizando o Weblate

2.3.1 Atualizações de imagem Docker

A imagem Docker oficial (veja *Instalando usando Docker*) possui todas as etapas de atualização integradas do Weblate. Normalmente não há nenhuma etapa manual necessária além de baixar a versão mais recente.

Ver também:

Atualizando o contêiner Docker

2.3.2 Instruções genéricas de atualização

Antes de atualizar, verifique os atuais *Requisitos de software*, pois eles podem ter sido alterados. Uma vez que todos os requisitos estejam instalados ou atualizados, ajuste 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 você esteja pulando algumas versões, siga as instruções para todas as versões que você está pulando na 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 você possa reverter o banco de dados caso a atualização falhe, veja *Fazendo backup e movendo 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"
```

Ou, se você quiser apenas obter a versão lançada mais recente:

```
pip install -U "Weblate[all]"
```

Se você não quiser instalar todas as dependências opcionais, faça:

```
pip install -U Weblate
```

Com um checkout do Git, você precisa obter o novo código-fonte e atualizar 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 você deseja.
4. Atualize o arquivo de configuração, consulte o `settings_example.py` ou *Instruções específicas de versão* para os passos necessários.
5. Atualize estrutura do banco de dados:

```
weblate migrate --noinput
```

6. Colete arquivos estáticos atualizados (veja *Executando servidor* e *Servindo arquivos estáticos*):

```
weblate collectstatic --noinput --clear
```

7. Compactar arquivos JavaScript e CSS (opcional, veja *Comprimindo os ativos do cliente*):

```
weblate compress
```

8. Se você está usando uma versão do Git, você também deve gerar novamente arquivos locais toda vez que estiver atualizando. Você pode fazer isso invocando:

```
weblate compilemessages
```

9. Verifique que sua configuração é sana (ver também *Configuração de produção*):

```
weblate check --deploy
```

10. Reinicie o worker do Celery (veja *Tarefas de fundo usando Celery*).

2.3.3 Instruções específicas de versão

Atualizar de 2.x

Se você está atualizando 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.

Ver também:

[Atualizar de 2.20 para 3.0 na documentação do Weblate 3.0](#)

Atualizar da 3.x

Se você está atualizando da versão 3.x, sempre atualize para a 4.0.4 ou 4.1.1 antes e depois continue atualizando na série 4.x. Atualizações pulando esse passo não são suportadas e não funcionarão corretamente.

Ver 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 suas configurações de acordo.
- Há novos formatos de arquivo, você pode querer incluí-los no caso de ter modificado o `WEBLATE_FORMATS`.
- Há novas verificações de qualidade, você pode querer 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`.
- A configuração `MT_DEEPL_API_VERSION` foi removida na versão 4.7. A tradução de máquina *DeepL* agora usa o novo `MT_DEEPL_API_URL`. Você pode precisar ajustar `MT_DEEPL_API_URL` para corresponder à sua assinatura.

Ver 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 aplicativos.
- 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 você depender de chaves nas exportações ou API.

- A configuração do Celery foi alterada para não usar mais a fila `memory`. Ajuste 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`). Você terá que configurá-lo antes de executar o Weblate.
- Os campos de nome de usuário e e-mail no banco de dados de usuários agora devem ser únicos e sem distinção de maiúsculo para minúsculo. Isso acidentalmente não foi reforçado com PostgreSQL.

Ver 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, você pode querer 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. Você precisará atualizar *Weblate Client* caso esteja usando.
- A migração do banco de dados para 4.3 pode demorar, dependendo do número de textos que você está traduzindo (espere cerca de uma hora de tempo de migração por 100.000 textos fonte).
- Há uma mudança em `django`: `INSTALLED_APPS`.
- Há uma nova configuração `SESSION_COOKIE_AGE_AUTHENTICATED` que complementa `SESSION_COOKIE_AGE`.
- Caso você esteja usando **hub** ou **lab** para integrar com GitHub ou GitLab, você 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 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 extra `skip_push`.

Ver 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 você esteja usando MySQL ou MariaDB, as versões mínimas necessárias aumentaram, veja *MySQL e MariaDB*.

Alterado na versão 4.4.1:

- *Gettext monolíngue* agora usa `msgid` e `msgctxt` quando presente. Isso mudará a identificação das sequências de tradução em tais arquivos, quebrando links para dados estendidos do Weblate, como capturas de tela ou estados de revisão. Certifique-se de confirmar as alterações pendentes em tais arquivos antes de atualizar e é recomendado forçar o carregamento do componente afetado usando `loadpo`.
- Incrementada a versão mínima necessária do `translate-toolkit` para resolver vários problemas de formato de arquivo.

Ver 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 você 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*.

Ver 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 arquivo, você pode querer 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 senhas.

Ver 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 suas configurações de acordo.
- A tradução de máquina *DeepL* agora tem uma configuração `MT_DEEPL_API_URL` genérica para se adaptar a diferentes modelos de assinatura de forma mais flexível. A configuração `MT_DEEPL_API_VERSION` não é mais usada.
- Django 3.2 agora é necessário.

Ver 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.

Ver 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.

Ver 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 demorar muito 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 contêiner Docker agora usa Django 4.0, consulte as alterações acima.

Ver também:

Instruções genéricas de atualização

Atualizar da 4.10 para 4.11

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- O Weblate agora requer Python 3.7 ou mais recente.
- A implementação de *Gerenciando controle de acesso por projeto* foi alterada, removendo o prefixo do projeto dos nomes dos grupos. Isso afeta os usuários 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`).

Ver também:

Instruções genéricas de atualização

Atualizar da 4.11 para 4.12

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- Não há etapas especiais necessárias.

Ver também:

Instruções genéricas de atualização

Atualizar da 4.12 para 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 desabilitar isso.
- A manipulação de contexto e local foi alterada para os formatos de arquivo *Arquivos RC do Windows*, *Arquivos HTML*, *Formato IDML* e *Arquivos texto*. Na maioria dos casos, o contexto agora é mostrado como local.
- Os serviços de tradução automática agora são configurados usando a interface do usuário, as configurações do arquivo de configuração serão importadas durante a migração do banco de dados.

Ver também:

Instruções genéricas de atualização

Atualizar da 4.13 para 4.14

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- As verificações de formatação Java agora correspondem aos marcadores GNU gettext. Os marcadores definidos no Weblate serão migrados automaticamente, mas scripts de terceiros precisam usar `java-printf-format` em vez de `java-format` e `java-format` em vez de `java-messageformat`.
- A dependência *jellyfish* foi substituída por *rapidfuzz*.
- **Alterado no 4.14.2:** Descontinuada a configuração insegura de chaves de API do serviço VCS por meio da configuração `_TOKEN/_USERNAME` em vez da lista `_CREDENTIALS`. No Docker, adicionar a diretiva `_HOST` correspondente. Por exemplo, confira `WEBLATE_GITHUB_HOST` e `GITHUB_CREDENTIALS`.

Ver também:

Instruções genéricas de atualização

Atualizar da 4.14 para 4.15

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- O Weblate agora requer a extensão `btree_gin` no PostgreSQL. O processo de migração irá instalá-lo se tiver privilégios suficientes. Veja *Criando um banco de dados no PostgreSQL* para configuração manual.
- A imagem do Docker não habilita mais o modo de depuração por padrão. Caso queira, habilite-o no ambiente usando `WEBLATE_DEBUG`.
- A migração do banco de dados pode levar horas em instâncias maiores devido à recriação de alguns dos índices.
- **Alterado no 4.15.1:** O valor padrão para `DEFAULT_PAGINATION_CLASS` nas configurações do framework foi alterado.

Ver também:

Instruções genéricas de atualização

Upgrade from 4.15 to 4.16

Siga as *Instruções genéricas de atualização* para realizar a atualização.

- Celery beat is now storing the tasks schedule in the database, `CELERY_BEAT_SCHEDULER` and `INSTALLED_APPS` need to be changed for that.
- The deprecated VCS setting for credentials is no longer supported, see *Atualizar da 4.13 para 4.14*.
- Upgrade of *django-crispy-forms* requires changes in `INSTALLED_APPS`.
- Integration of *django-cors-headers* requires changes in `INSTALLED_APPS` and `MIDDLEWARE`.

Ver 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 você ainda esteja executando uma versão mais antiga, migre para o Python 3 primeiro na versão existente e atualize mais tarde. Consulte *Atualizando do Python 2 para o Python 3* na documentação do Weblate 3.11.1.

2.3.5 Migrando de outros bancos de dados para o PostgreSQL

Se você estiver executando o Weblate em outro banco de dados que não o PostgreSQL, deve considerar a migração para o PostgreSQL, pois o Weblate tem melhor desempenho com ele. As etapas a seguir o orientarão na migração de seus dados entre os bancos de dados. Lembre-se de parar os servidores web e Celery antes da migração, caso contrário, você pode acabar com dados inconsistentes.

Criando um banco de dados no PostgreSQL

Geralmente é uma boa ideia executar o Weblate em um banco de dados separado e separar a conta do usuário:

```
# If PostgreSQL was not installed before, set the main password
sudo -u postgres psql postgres -c "\password postgres"

# Create a database user called "weblate"
sudo -u postgres createuser -D -P weblate

# Create the database "weblate" owned by "weblate"
sudo -u postgres createdb -E UTF8 -O weblate weblate
```

Migrando usando 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 você pode querer usar o pgloader, veja *Migrando para PostgreSQL usando pgloader*.

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",
```

(continua na próxima página)

(continuação da página anterior)

```

# Database password
"PASSWORD": "password",
# Set to empty string for localhost
"HOST": "database.example.com",
# Set to empty string for default
"PORT": "",
# Additional database options
"OPTIONS": {
    # In case of using an older MySQL server, which has MyISAM as a
↪default storage
    # 'init_command': 'SET storage_engine=INNODB',
    # Uncomment for MySQL older than 5.7:
    # 'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
    # If your server supports it, see the Unicode issues above
    "charset": "utf8mb4",
    # Change connection timeout in case you get MySQL gone away error:
    "connect_timeout": 28800,
},
},
"postgresql": {
    # Database engine
    "ENGINE": "django.db.backends.postgresql",
    # Database name
    "NAME": "weblate",
    # Database user
    "USER": "weblate",
    # Database password
    "PASSWORD": "password",
    # Set to empty string for localhost
    "HOST": "database.example.com",
    # Set to empty string for default
    "PORT": "",
},
}

```

2. 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 para o 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 legado.

O Weblate agora deve estar pronto para ser executado a partir do banco de dados PostgreSQL.

Migrando para PostgreSQL usando pgloader

O `pgloader` é uma ferramenta de migração genérica para migrar dados para PostgreSQL. Você pode usá-lo para migrar o banco de dados Weblate.

1. Ajuste 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 você pode querer 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 usuário do Pootle. Você pode descartar os usuários do Pootle e importá-los usando `importusers`.

2.4 Fazendo backup e movendo o Weblate

2.4.1 Backups de nível de projeto

Novo na versão 4.14.

Aviso: A restauração de backups só é suportada quando se utiliza PostgreSQL ou MariaDB 10.5+ como banco de dados.

O projeto faz backup de todo o conteúdo da tradução da Weblate (projeto, componentes, traduções, comentários de textos, sugestões ou verificações). É indicado para transferir um projeto para outra instância do Weblate.

Você pode realizar um backup do projeto em *Gerenciar* ↓ *Backups*. O backup pode ser restaurado ao criar um projeto (veja *Adicionando projetos e componentes de tradução*).

Os backups no momento não incluem o histórico e as informações de controle de acesso.

Os comentários e sugestões são acompanhados pelo nome do usuário que os criou. Após a importação, ele é atribuído a um usuário correspondente. Se não houver nenhum usuário com tal nome de usuário, ele é atribuído a um usuário anônimo.

Os backups gerados são mantidos no servidor conforme configurado por `PROJECT_BACKUP_KEEP_DAYS` e `PROJECT_BACKUP_KEEP_COUNT` (o padrão é manter no máximo 3 backups por 30 dias).

2.4.2 Backup automatizado usando BorgBackup

Novo na versão 3.9.

O Weblate tem suporte embutido para criação de backups de serviços usando [BorgBackup](#). Borg cria backups criptografados eficazes em termos de espaço que podem ser armazenados com segurança na nuvem. Os backups podem ser controlados na interface de gerenciamento da aba *Backups*.

Alterado na versão 4.4.1: 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

W Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾

⚙️

+

🌐

⋮

Manage / Backups

Backup process triggered

Weblate status

Backups

Translation memory

Performance report

SSH keys

Alerts

Repositories

Users

Teams

Appearance

Tools

Automatic suggestions

Billing

Backup service: /tmp/tmp8o1nk_tnweblate

Backup service credentials

March 1, 2023

Backup repository

/tmp/tmp8o1nk_tnweblate

Passphrase

2zwuwVXXKVI5dr\$pyvuga fL6FwBdzr6CYBTUXwT\$z^OmHO&gc^ (

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

March 1, 2023

Backup performed

March 1, 2023

Repository initialization

March 1, 2023

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 4.16 About Weblate Legal Contact Documentation Donate to Weblate

240

Capítulo 2. Documentação de administrador

Chave de criptografia do Borg

BorgBackup cria backups criptografados e você não conseguiria restaurá-los sem a senha. A senha é gerada ao adicionar novo serviço de backup e você deve copiá-lo e mantê-lo em um lugar seguro.

Se você estiver usando *Armazenamento de backup provisionado do Weblate*, faça backup da sua chave SSH privada também, pois ela é usada para acessar seus backups.

Ver também:

`borg init`

Personalizando o backup

- O backup do banco de dados pode ser configurado via `DATABASE_BACKUP`.
- A criação do backup pode ser personalizada usando `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 gerenciamento, veja *Integrando suporte*.
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 aba *Backups*.
5. Faça backup de 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 registro.

2.4.4 Usando armazenamento de backup personalizado

Você também pode usar seu próprio armazenamento para backups. SSH pode ser usado para armazenar backups no destino remoto, o servidor de destino precisa do **BorgBackup** instalado.

Ver também:

[General](#) na documentação do Borg

Sistema de arquivos local

Recomenda-se especificar o caminho absoluto para o backup local, por exemplo `/caminho/para/backup`. O diretório deve poder ser escrito pelo usuário executando o Weblate (veja *Permissões do sistema de arquivos*). 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 contêiner Weblate. Caso contrário, os backups serão descartados pelo Docker na reinicialização do seu contêiner.

Uma opção é colocar backups em um volume existente. Por exemplo, `/app/data/borgbackup`. Este é um volume existente no contêiner.

Você também pode adicionar um novo contêiner para os backups no arquivo de composição do Docker, por exemplo, usando `/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, você terá que instalar o [BorgBackup](#) em outro servidor que seja acessível para sua implantação de Weblate via SSH usando a chave SSH do Weblate:

1. Prepare um servidor onde seus backups serão armazenados.
2. Instale o servidor SSH nele (você o receberá 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 usuário existente ou crie um novo que será usado para backup.
5. Adicione a chave SSH do Weblate ao usuário para que o Weblate possa usar o SSH para o servidor sem uma senha (veja [Chave SSH do Weblate](#)).
6. Configure o local de backup no Weblate como `usuário@host:/caminho/para/backups` ou `usuário@host:/caminho/para/backups`.

Dica: [Armazenamento de backup provisionado do Weblate](#) fornece backups remotos automatizados sem qualquer esforço.

Ver também:

[Chave SSH do Weblate](#), [General](#)

2.4.5 Restaurando do BorgBackup

1. Restaure o acesso ao repositório de backup e prepare sua senha de backup.
2. Liste todos os backups no servidor usando `borg list REPOSITÓRIO`.
3. Restaure o backup desejado para o diretório atual usando `borg extract REPOSITÓRIO::PACOTE`.
4. Restaure o banco de dados a partir do despejo de SQL colocado no diretório `backup` no diretório de dados do Weblate (veja [Dados despejados para os backups](#)).
5. Copie a configuração do Weblate (`backups/settings.py`, veja [Dados despejados para os backups](#)) até o local correto, veja [Ajustando a configuração](#).

Ao usar o contêiner Docker, o arquivo de configurações já está incluído no contêiner e você deve restaurar as variáveis de ambiente originais. O arquivo `environment.yml` pode ajudá-lo com isso (veja [Dados despejados para os backups](#)).

6. Copie todo o diretório de dados restaurados para o local configurado por `DATA_DIR`.

Ao usar contêiner do Docker, coloca os dados em um volume de dados, veja [Volumes de contêiner Docker](#).

Por favor, certifique-se de que os arquivos possuem a propriedade e permissões corretas, veja [Permissões do sistema de arquivos](#).

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:
```

Ver também:

[borg list](#), [borg extract](#)

2.4.6 Backup manual

Dependendo do que você deseja salvar, faça backup do tipo de dados que o Weblate armazena em cada lugar.

Dica: Se você estiver fazendo os backups manualmente, você pode querer silenciar os avisos do Weblate sobre a falta de backups adicionando `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 salvar um despejo do banco de dados usando ferramentas nativas, tais como `pg_dump` ou `mysqldump`. Esta abordagem normalmente tem um desempenho melhor do que o backup do Django e restaura tabelas completas com todos os seus dados.

Você pode restaurar esse backup em uma 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, você pode fazer backup de seu banco de dados utilizando o comando `dumpdata` do Django. Dessa forma o backup é agnóstico de banco de dados e pode ser usado caso você queira alterar o backend do banco de dados.

Antes de restaurar o banco de dados, você precisa estar usando 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 você acabaria corrompendo os dados de alguma forma. Depois de instalar a mesma versão, execute todas as migrações do banco de dados usando `migrate`.

Depois disso, algumas entradas já serão criadas no banco de dados e você as terá no backup do banco de dados também. A abordagem recomendada é excluir essas entradas manualmente usando o shell de gerenciamento (veja *Invocando comandos de gerenciamento*):

```
weblate shell
>>> from weblate.auth.models import User
>>> User.objects.get(username='anonymous').delete()
```

Arquivos

Se você tiver espaço de backup suficiente, basta fazer backup de todo o `DATA_DIR`. Esta é uma aposta segura, mesmo que inclua alguns arquivos que você não quer. As seções a seguir descrevem o que você deve fazer backup e o que você pode pular em detalhes.

Dados despejados para os 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 você pode incluir esses arquivos para backups mais completos. Os arquivos são atualizados diariamente (requer um servidor de “beats” do Celery em execução, consulte *Tarefas de fundo usando Celery*). Atualmente, isso inclui:

- Configurações do Weblate como `settings.py` (existe também a versão expandida em `settings-expanded.py`).
- Backup de banco de dados PostgreSQL como `database.sql`.
- 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 usando `DATABASE_BACKUP`.

Para restaurar o backup do banco de dados, carregue-o usando ferramentas de banco de dados, por exemplo:

```
psql --file=database.sql weblate
```

Repositórios de controle de versão

Armazenados em `DATA_DIR/vcs`.

Os repositórios de controle de versão contêm uma cópia de seus repositórios upstream com alterações do Weblate. Se você tiver o *Push ao fazer commit* 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 você está usando chaves SSH ou GPG geradas pelo Weblate, você deve fazer backup destes locais. Caso contrário, você vai perder as chaves privadas e você terá que gerar novamente as novas.

Arquivos enviados pelo usuário

Armazenados em `DATA_DIR/media`.

Você deve fazer o backup de todos os arquivos enviados pelo usuário (por exemplo, *Contexto visual para textos*).

Tarefas do Celery

A fila de tarefas do Celery pode conter algumas informações, mas geralmente não é necessária para um backup. No máximo, você 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.

Ver também:

Tarefas de fundo usando Celery

Linha de comando para backup manual

Usando uma tarefa de cron, você 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
```

O texto entre as aspas após `XZ_OPT` permite que você escolha suas opções do xz, por exemplo, a quantidade de memória utilizada para compressão; veja <https://linux.die.net/man/1/xz>

Você pode ajustar a lista de pastas e arquivos às suas necessidades. Para evitar salvar a memória de tradução (na pasta backups), você 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 Restaurando backup manual

1. Restaure todos os dados dos quais você tenha feito backup.
2. Atualize todos repositórios usando o `updategit`.

```
weblate updategit --all
```

2.4.8 Movendo uma instalação do Weblate

Realoque a instalação de um sistema diferente, seguindo as instruções de backup e restauração acima.

Ver também:

Atualização de Python 2 para Python 3, Migrando de outros bancos de dados para o PostgreSQL

2.5 Autenticação

2.5.1 Registro de usuário

A configuração padrão para Weblate é usar `python-social-auth`, um formulário no site para lidar com o registro de novos usuários. Depois de confirmar seu e-mail, um novo usuário pode contribuir ou autenticar usando um dos serviços de terceiros.

Você também pode desativar o registro de novos usuários usando `REGISTRATION_OPEN`.

As tentativas de autenticação estão sujeitas a *Limitação de taxa*.

2.5.2 Backends de autenticação

A solução embutida do Django é utilizada para autenticação, incluindo várias opções sociais para o fazer. Utilizando-a, você pode importar o banco de dados de usuários de outros projetos baseados no Django (veja *Migrando de Pootle*).

Django pode, adicionalmente, ser configurado para autenticar em outros meios também.

Ver também:

Configurações de autenticação descreve como configurar a autenticação na imagem oficial do Docker.

2.5.3 Autenticação social

Graças ao [Welcome to Python Social Auth's documentation!](#), o Weblate tem suporte a autenticação utilizando muitos serviços de terceiros, tais como GitLab, Ubuntu, Fedora etc.

Por favor, verifique sua documentação para as instruções de configuração genéricas em [Django Framework](#).

Nota: Por padrão, o Weblate conta com serviços de autenticação de terceiros para fornecer um endereço de e-mail validado. Se alguns dos serviços que você deseja usar não suportarem isso, por favor, aplique a validação de e-mail no lado Weblate configurando `FORCE_EMAIL_VALIDATION` para eles. Por exemplo:

```
SOCIAL_AUTH_OPENSUSE_FORCE_EMAIL_VALIDATION = True
```

Ver também:

[Pipeline](#)

Permitir backends individuais é bastante fácil, é apenas uma questão de adicionar uma entrada à configuração `AUTHENTICATION_BACKENDS` e possivelmente adicionar chaves necessárias para um determinado método de autenticação. Por favor, note que alguns backends não fornecem e-mails do usuário por padrão, você tem que solicitá-lo explicitamente, caso contrário o Weblate não será capaz de pagar corretamente as contribuições que os usuários fazem.

Dica: A maioria dos backends de autenticação exige HTTPS. Assim que o HTTPS estiver habilitado em seu servidor web, configure o Weblate para relatá-lo corretamente usando `ENABLE_HTTPS` ou, no contêiner Docker, `WEBLATE_ENABLE_HTTPS`.

Ver também:

[Backend de Python Social Auth](#)

Autenticação por OpenID

Para serviços baseados em OpenID, geralmente é apenas uma questão de habilitá-los. A seção a seguir permite a autenticação OpenID para OpenSUSE, Fedora e Ubuntu:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.suse.OpenSUSEOpenId",
    "social_core.backends.ubuntu.UbuntuOpenId",
    "social_core.backends.fedora.FedoraOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

Ver também:

[OpenID](#)

Autenticação por GitHub

Você precisa registrar um aplicativo OAuth no GitHub e, em seguida, dizer ao Weblate todos os seus segredos:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.github.GithubOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = "GitHub Client ID"
SOCIAL_AUTH_GITHUB_SECRET = "GitHub Client Secret"
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]
```

O GitHub deve ser configurado para ter URL de um retorno de chamada como `https://example.com/accounts/complete/github/`.

Existem backends de autenticação semelhantes para o GitHub para Organizações e o GitHub para Equipes. Suas configurações 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`. Seus URLs de retorno 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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[GitHub](#)

Autenticação por Bitbucket

Você precisa registrar um aplicativo no Bitbucket e, em seguida, dizer ao Weblate todos os seus segredos:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.bitbucket.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 você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[Bitbucket](#)

Google OAuth 2

Para usar o OAuth 2 do Google, você precisa se registrar em um aplicativo em <https://console.developers.google.com/> e ativar a API do Google+.

A URL de redirecionamento é `https://SERVIDOR/WEBLATE/accounts/complete/google-oauth2/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.google.GoogleOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = "Client ID"
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = "Client secret"
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[Google](#)

OAuth 2 do Facebook

Como de costume com os serviços OAuth 2, você precisa registrar seu aplicativo no Facebook. Uma vez feito isso, você pode configurar o Weblate para usá-lo:

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/facebook/`

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.facebook.FacebookOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_FACEBOOK_KEY = "key"
SOCIAL_AUTH_FACEBOOK_SECRET = "secret"
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

Facebook

OAuth 2 do GitLab

Para usar o OAuth 2 do GitLab, você precisa registrar um aplicativo em <https://gitlab.com/profile/applications>.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/gitlab/` e garantir que você marque o escopo `read_user`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.gitlab.GitLabOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_GITLAB_KEY = "Application ID"
SOCIAL_AUTH_GITLAB_SECRET = "Secret"
SOCIAL_AUTH_GITLAB_SCOPE = ["read_user"]

# If you are using your own GitLab
# SOCIAL_AUTH_GITLAB_API_URL = 'https://gitlab.example.com/'
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

GitLab

Active Directory do Microsoft Azure

Weblate pode ser configurado para usar inquilinos comuns ou específicos para autenticação.

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/azuread-oauth2/` para autenticação comum e `https://SERVIDOR WEBLATE/accounts/complete/azuread-tenant-oauth2/` para autenticação específica do inquilino.

```
# Azure AD common

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread.AzureADOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET = ""
```

```
# Azure AD Tenant

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.azuread_tenant.AzureADTenantOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# OAuth2 keys
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY = ""
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET = ""
# Tenant ID
SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TENANT_ID = ""
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

Microsoft Azure Active Directory

Slack

Para usar o OAuth 2 do Slack, você precisa cadastrar um aplicativo em [<https://api.slack.com/apps>](https://api.slack.com/apps).

A URL de redirecionamento é `https://SERVIDOR WEBLATE/accounts/complete/slack/`.

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.slack.SlackOAuth2",
    "social_core.backends.email.EmailAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SLACK_KEY = ""
SOCIAL_AUTH_SLACK_SECRET = ""
```

Nota: O Weblate fornecia URL de retorno de chamada durante a autenticação inclui domínio configurado. No caso de você obter erros sobre incompatibilidade de URL, você pode querer corrigir isso, consulte [Definir domínio correto do site](#).

Ver também:

[Slack](#)

Substituindo nomes e ícones de métodos de autenticação

Você pode substituir o nome de exibição do método de autenticação e o ícone usando 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"
```

Desativando autenticação por senha

Autenticação por e-mail e senha pode ser desativada através da remoção de `social_core.backends.email.EmailAuth` de `AUTHENTICATION_BACKENDS`. Mantenha sempre `weblate.accounts.auth.WeblateUserBackend` lá, ele é necessário para a funcionalidade central do Weblate.

Desabilitar a autenticação por e-mail desabilitará todas as funcionalidades relacionadas a e-mail – convite a usuários ou recurso de redefinição de senha.

Dica: Você ainda pode usar autenticação por senha para a interface administrativa, para usuários que você cria manualmente lá. Basta navegar para `/admin/login/`.

Por exemplo, a autenticação usando apenas o provedor Open ID do openSUSE pode ser alcançada usando o seguinte:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.suse.OpenSUSEOpenId",
    "weblate.accounts.auth.WeblateUserBackend",
)
```

2.5.4 Autenticação por senha

O `settings.py` padrão vem com um razoável conjunto de `AUTH_PASSWORD_VALIDATORS`:

- As senhas não podem ser muito similares às suas outras informações pessoais.
- As senhas devem conter no mínimo 10 caracteres.
- As senhas não podem ser uma senha comumente usada.
- As senhas não podem ser inteiramente numéricas.
- As senhas não podem consistir em um único caractere ou apenas espaço em branco.
- As senhas não podem corresponder a uma senha que você usou no passado.

Você pode personalizar esta configuração para corresponder à sua política de senha.

Além disso, você também pode instalar o `django-zxcvbn-password` o que dá bastante estimativas realistas de senha dificuldade e permite rejeitar senhas abaixo de um determinado limite.

2.5.5 Autenticação por SAML

Novo na versão 4.1.1.

Siga as instruções do Python Social Auth para configuração. Diferenças notáveis:

- Weblate tem suporte a único IDP que tem de ser chamado de weblate em SOCIAL_AUTH_SAML_ENABLED_IDPS.
- A URL de metadados XML de SAML é /accounts/metadata/saml/.
- As configurações a seguir são preenchidas automaticamente: SOCIAL_AUTH_SAML_SP_ENTITY_ID, SOCIAL_AUTH_SAML_TECHNICAL_CONTACT, SOCIAL_AUTH_SAML_SUPPORT_CONTACT

Exemplo de configuração:

```
# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    "social_core.backends.saml.SAMLAuth",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Social auth backends setup
SOCIAL_AUTH_SAML_SP_ENTITY_ID = f"https://{SITE_DOMAIN}/accounts/metadata/saml/"
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT = "-----BEGIN CERTIFICATE-----"
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY = "-----BEGIN PRIVATE KEY-----"
SOCIAL_AUTH_SAML_ENABLED_IDPS = {
    "weblate": {
        "entity_id": "https://idp.testshib.org/idp/shibboleth",
        "url": "https://idp.testshib.org/idp/profile/SAML2/Redirect/SSO",
        "x509cert": "MIIEDjCCAvagAwIBAgIBADA ... 8Bbn1+ev0peYzxFyF5sQA==",
        "attr_name": "full_name",
        "attr_username": "username",
        "attr_email": "email",
    }
}
SOCIAL_AUTH_SAML_ORG_INFO = {
    "en-US": {
        "name": "example",
        "displayname": "Example Inc.",
        "url": "http://example.com"
    }
}
SOCIAL_AUTH_SAML_TECHNICAL_CONTACT = {
    "givenName": "Tech Gal",
    "emailAddress": "technical@example.com"
}
SOCIAL_AUTH_SAML_SUPPORT_CONTACT = {
    "givenName": "Support Guy",
    "emailAddress": "support@example.com"
}
```

A configuração padrão extrai detalhes do usuário dos seguintes atributos, configure 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
Sobrenome	urn:oid:2.5.4.4
E-mail	urn:oid:0.9.2342.19200300.100.1.3
Nome de usuário	urn:oid:0.9.2342.19200300.100.1.1

Dica: O exemplo acima e a imagem do Docker definem um IDP chamado `weblate`. Pode ser preciso configurar este texto como *Relay* em seu IDP.

Ver também:

Configurando SAML no Docker, SAML

2.5.6 Autenticação por LDAP

A autenticação por LDAP pode ser melhor alcançada utilizando o pacote *django-auth-ldap*. Você pode instalá-lo através dos meios habituais:

```
# Using PyPI
pip install django-auth-ldap>=1.3.0

# Using apt-get
apt-get install python-django-auth-ldap
```

Dica: Este pacote está incluído no contêiner Docker, veja *Instalando usando Docker*.

Nota: Há algumas incompatibilidades no módulo Python LDAP 3.1.0, o que pode impedir você de usar essa versão. Se você obter o erro `AttributeError: 'module' object has no attribute '_trace_level'`, fazer o downgrade para python-ldap 3.0.0 pode ajudar.

Uma vez que você tenha o pacote instalado, você pode conectá-lo à autenticação do Django:

```
# Add LDAP backed, keep Django one if you want to be able to sign in
# even without LDAP for admin account
AUTHENTICATION_BACKENDS = (
    "django_auth_ldap.backend.LDAPBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# LDAP server address
AUTH_LDAP_SERVER_URI = "ldaps://ldap.example.net"

# DN to use for authentication
AUTH_LDAP_USER_DN_TEMPLATE = "cn=%(user)s,o=Example"
# Depending on your LDAP server, you might use a different DN
# like:
# AUTH_LDAP_USER_DN_TEMPLATE = 'ou=users,dc=example,dc=com'

# List of attributes to import from LDAP upon sign in
# Weblate stores full name of the user in the full_name attribute
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "name",
    # Use the following if your LDAP server does not have full name
    # Weblate will merge them later
    # 'first_name': 'givenName',
    # 'last_name': 'sn',
    # Email is required for Weblate (used in VCS commits)
    "email": "mail",
}

# Hide the registration form
REGISTRATION_OPEN = False
```

Nota: Você deve remover 'social_core.backends.email.EmailAuth' da configuração `AUTHENTICATION_BACKENDS`; caso contrário, os usuários poderão definir sua senha no Weblate e autenticar usando isso. Manter 'weblate.accounts.auth.WeblateUserBackend' ainda é necessário para fazer permissões e facilitar usuários anônimos. Ele também permitirá que você faça login usando uma conta administrativa local, se você a criou (por exemplo, usando `createadmin`).

Usando senha associada

Se você não puder usar associação direta para autenticação, você precisará usar a pesquisa e fornecer um usuário para associar à pesquisa. Por exemplo:

```
import ldap
from django_auth_ldap.config import LDAPSearch

AUTH_LDAP_BIND_DN = ""
AUTH_LDAP_BIND_PASSWORD = ""
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "ou=users,dc=example,dc=com", ldap.SCOPE_SUBTREE, "(uid=%(user)s)"
)
```

Integração com Active Directory

```
import ldap
from django_auth_ldap.config import LDAPSearch, NestedActiveDirectoryGroupType

AUTH_LDAP_BIND_DN = "CN=ldap,CN=Users,DC=example,DC=com"
AUTH_LDAP_BIND_PASSWORD = "password"

# User and group search objects and types
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "CN=Users,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(sAMAccountName=%(user)s)"
)

# Make selected group a superuser in Weblate
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    # is_superuser means user has all permissions
    "is_superuser": "CN=weblate_AdminUsers,OU=Groups,DC=example,DC=com",
}

# Map groups from AD to Weblate
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "OU=Groups,DC=example,DC=com", ldap.SCOPE_SUBTREE, "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedActiveDirectoryGroupType()
AUTH_LDAP_FIND_GROUP_PERMS = True

# Optionally enable group mirroring from LDAP to Weblate
# AUTH_LDAP_MIRROR_GROUPS = True
```

Ver também:

Django Authentication Using LDAP, Authentication

2.5.7 Autenticação por CAS

A autenticação por CAS pode ser alcançada usando um pacote como o *django-cas-ng*.

O primeiro passo é divulgar o campo de e-mail do usuário via CAS. Isso tem que ser configurado no próprio servidor CAS, e requer que você execute pelo menos CAS v2, já que o CAS v1 não tem suporte a atributos.

O segundo passo é atualizar o Weblate para utilizar o seu servidor CAS e os seus atributos.

Para instalar *django-cas-ng*:

```
pip install django-cas-ng
```

Uma vez que o pacote instalado, você pode conectá-lo ao sistema de autenticação do Django modificando o arquivo `settings.py`:

```
# Add CAS backed, keep the Django one if you want to be able to sign in
# even without LDAP for the admin account
AUTHENTICATION_BACKENDS = (
    "django_cas_ng.backends.CASBackend",
    "weblate.accounts.auth.WeblateUserBackend",
)

# CAS server address
CAS_SERVER_URL = "https://cas.example.net/cas/"

# Add django_cas_ng somewhere in the list of INSTALLED_APPS
INSTALLED_APPS = (... , "django_cas_ng")
```

Finalmente, um sinal pode ser usado para mapear o campo de e-mail para o objeto do usuário. Para que isso funcione, você tem que importar o sinal do pacote *django-cas-ng* e conectar seu código com este sinal. Fazer isso em configurações de arquivo pode causar problemas, portanto, é sugerido colocá-lo:

- No método `django.apps.AppConfig.ready()` da configuração do seu aplicativo
- No arquivo `urls.py` do projeto (quando não há modelos)

```
from django_cas_ng.signals import cas_user_authenticated
from django.dispatch import receiver

@receiver(cas_user_authenticated)
def update_user_email_address(sender, user=None, attributes=None, **kwargs):
    # If your CAS server does not always include the email attribute
    # you can wrap the next two lines of code in a try/catch block.
    user.email = attributes["email"]
    user.save()
```

Ver também:

Django CAS NG

2.5.8 Configurando autenticação por Django de terceiros

Geralmente, qualquer plugin de autenticação Django deve funcionar com Weblate. Basta seguir as instruções do plugin, lembrando de manter o backend do usuário Weblate instalado.

Ver também:

Autenticação por LDAP, Autenticação por CAS

Normalmente, a instalação consiste em adicionar uma autenticação de backend a `AUTHENTICATION_BACKENDS` e a instalar um aplicativo de autenticação (se houver) no `INSTALLED_APPS`:

```
AUTHENTICATION_BACKENDS = (  
    # Add authentication backend here  
    "weblate.accounts.auth.WeblateUserBackend",  
)  
  
INSTALLED_APPS += (  
    # Install authentication app here  
)
```

2.6 Controle de acesso

O Weblate vem com um sistema de privilégios fino para atribuir permissões ao usuário para toda a instância ou em um escopo limitado.

Alterado na versão 3.0: Antes do Weblate 3.0, o sistema de privilégios era baseado apenas no sistema de privilégios do Django, mas é agora especificamente construído para Weblate. Se for usado algo mais antigo, consulte a documentação para a versão específica que você está usando.

2.6.1 Controle de acesso personalizado

Se você não está administrando toda a instalação do Weblate e apenas tem acesso para gerenciar certos projetos (como no [Hosted Weblate](#)), suas opções de gerenciamento de controle de acesso são limitadas às seguintes configurações. Se você não precisa de nenhuma configuração complexa, elas são suficientes para você.

Controle de acesso do projeto

Nota: Este recurso está indisponível para projetos que executam o plan Libre no Hosted Weblate.

Você pode limitar o acesso do usuário a projetos individuais selecionando uma configuração diferente de *Controle de acesso*. As opções disponíveis são:

Público

Visível publicamente, traduzível para todos os usuários autenticados.

Protegido

Publicamente visível, mas somente traduzível por usuários selecionados.

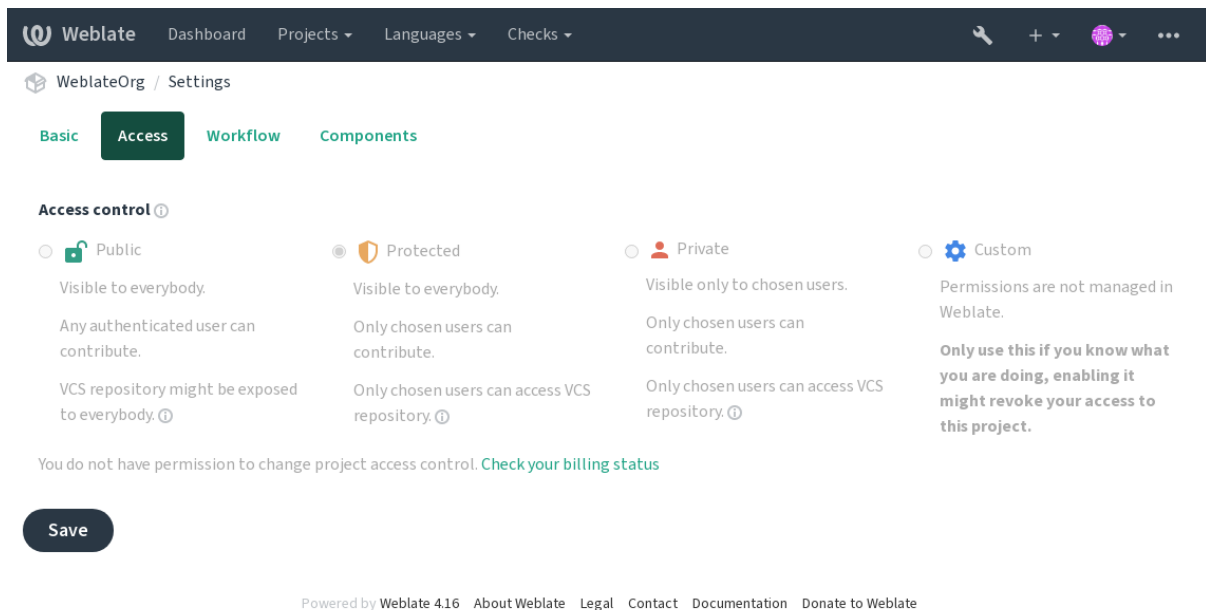
Privado

Visível e traduzível apenas por usuários selecionados.

Personalizado

Os recursos de *Gerenciamento de usuários* serão desabilitados; por padrão, todos os usuários estão proibidos de realizar qualquer ação no projeto. Você terá que configurar todas as permissões usando *Controle de acesso personalizado*.

O *Controle de acesso* pode ser alterado na aba *Acesso* da configuração (*Gerenciar* ↓ *Configurações*) de cada projeto.



O valor padrão pode ser alterado por `DEFAULT_ACCESS_CONTROL`.

Nota: Mesmo para projetos *Privados*, algumas informações sobre o seu projeto serão expostas: estatísticas e resumo de linguagem para toda a instância incluirão contagens para todos os projetos, apesar da configuração de controle de acesso. O nome do seu projeto e outras informações não podem ser revelados por meio disso.

Nota: O conjunto real de permissões disponíveis para usuários por padrão em projetos *Públicos*, *Protegidos* e *Privados* pode ser redefinido pelo administrador da instância de Weblate usando *configurações personalizadas*.

Ver também:

[Controle de acesso](#)

Gerenciando controle de acesso por projeto

Os usuários com o privilégio *Gerenciar acesso ao projeto* (consulte [Lista de privilégios e funções embutidas](#)) podem gerenciar usuários em projetos adicionando-os às equipes. A coleção inicial de equipes é fornecida pelo Weblate, mas outros adicionais podem ser definidos fornecendo um controle de acesso mais refinado. Você pode limitar as equipes a idiomas e atribuir-lhes funções de acesso designadas (consulte [Lista de privilégios e funções embutidas](#)).

As seguintes equipes 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 textos fonte (se permitido nas *configurações do projeto*) e informações de textos fonte.

Idiomas

Pode gerenciar idiomas traduzidos (adicionar ou remover traduções).

Glossário

Pode gerenciar glossário (adicionar ou remover entradas, e também enviar).

Memória

Pode gerenciar memória de tradução.

Capturas de tela

Pode gerenciar capturas de tela (adicioná-las ou removê-las e associá-las a textos fonte).

Tradução automática

Pode usar tradução automática.

VCS

Pode gerenciar VCS e acessar o repositório exportado.

Cobrança

Pode acessar informações de cobrança e configurações (consulte [Cobrança](#)).

Users

Username	Full name	E-mail	Last sign in	Teams
testuser	Weblate Test	weblate@example.org	16 seconds ago	Translate

Once all its permissions are removed, the user will be removed from the project.

Add a user

User to add

Please type in an existing Weblate account name or e-mail address.

Add

Block user

User to block

Please type in an existing Weblate account name or e-mail address.

Block duration

Block the user until I unblock

Block

Invite new user

E-mail

Username

Username may only contain letters, numbers or the following characters: @ . + - _

Full name

Invite

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Esses recursos estão disponíveis na página *Controle de acesso*, que pode ser acessada no projeto a partir do menu *Gerenciar* ↓ *Usuários*.

Administradores de equipe

Novo na versão 4.15.

Cada equipe pode ter um administrador de equipe, que pode adicionar e remover usuários dentro da equipe. Isso é útil no caso de você querer construir equipes autogovernadas.

Novo convite de usuário

Além de adicionar um usuário existente ao projeto, também é possível convidar novos. Qualquer novo usuário será criado imediatamente, mas a conta permanecerá inativa até que você entre com um link no convite enviado por e-mail. Não é necessário ter nenhum privilégio de todo o site para fazer isso, a permissão de gerenciamento de acesso no escopo do projeto (por exemplo, uma associação na equipe *Administração*) seria suficiente.

Dica: Se o usuário convidado perdeu a validade do convite, ele pode definir sua senha usando o endereço de e-mail convidado no formulário de redefinição de senha, pois a conta já foi criada.

Novo na versão 3.11: É possível reenviar o e-mail para convites de usuário (invalidando qualquer convite enviado anteriormente).

O mesmo tipo de convite está disponível em todo o site a partir da *interface de gerenciamento* na aba *Usuários*.

Bloqueando usuários

Novo na versão 4.7.

No caso de alguns usuários se comportarem mal em seu projeto, você tem a opção de bloqueá-los de contribuir. O usuário bloqueado ainda poderá ver o projeto se tiver permissão para isso, mas não poderá contribuir.

Gerenciamento de permissões por projeto

Você pode definir seus projetos para *Protegido* ou *Privado* e *gerenciar usuários* por projeto na interface de usuário do Weblate.

Por padrão, isso impede que o Weblate conceda acesso fornecido por *equipes padrão* *Usuários* e *Visualizadores* devido à própria configuração dessas equipes. Isso não impede que você conceda permissões a esses projetos em todo o site, alterando equipes padrão, criando uma nova ou criando configurações personalizadas adicionais para componentes individuais, conforme descrito em *Controle de acesso personalizado* abaixo.

Um dos principais benefícios de gerenciar permissões por meio da interface de usuário do Weblate é que você pode delegá-las a outros usuários sem dar a eles o privilégio de superusuário. Para isso, adicione-os à equipe *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 equipes e funções, onde as funções de definir um conjunto de permissões, e equipes vinculam-nas para os usuários e traduções. Veja *Usuários, papéis, equipes e permissões* para mais detalhes.

Os recursos mais poderosos do sistema de controle de acesso do Weblate por enquanto estão disponíveis apenas através da *interface administrativa do Django*. Você pode usá-la para gerenciar as permissões de qualquer projeto. Você não precisa necessariamente definir o *controle de acesso* para *Personalizado* para utilizá-lo. No entanto, você deve ter privilégios de superusuário para usá-lo.

Se você não está interessado em detalhes de implementação, e apenas deseja criar uma configuração simples o suficiente com base nos padrões, ou não tem um acesso de todo o site para toda a instalação do Weblate (como em [Hosted Weblate](#)), consulte a seção *Controle de acesso personalizado*.

Configurações comuns

Esta seção contém uma visão geral de algumas configurações comuns nas quais você pode estar interessado.

Gerenciamento de permissões para todo o site

Para gerenciar permissões para uma instância inteira de uma vez, adicione usuários às *equipes padrão*:

- *Usuários* (isso é feito por padrão pela *associação de equipe automática*).
- *Revisores* (se você estiver usando *fluxo de trabalho de revisão* com revisores dedicados).
- *Gestores* (se você deseja delegar a maioria das operações de gestão a outra pessoa).

Você deve manter todos os projetos configurados como *Públicos* (veja *Controle de acesso do projeto*), caso contrário as permissões de todo o site fornecidas pela participação nas equipes *Usuários* e *Revisores* não terão nenhum efeito.

Você também pode conceder algumas permissões adicionais de sua escolha às equipes padrão. Por exemplo, você pode querer dar permissão para gerenciar capturas de tela para todos os *Usuários*.

Você também pode definir algumas novas equipes personalizadas. Se você deseja continuar gerenciando suas permissões em todo o site para essas equipes, escolha um valor apropriado para *Seleção de projeto* (por exemplo *Todos os projetos* ou *Todos os projetos públicos*).

Permissões personalizadas a idiomas, componentes ou projetos

Você pode criar suas próprias equipes dedicadas para gerenciar permissões para objetos distintos, como linguagens, componentes e projetos. Embora essas equipes só possam conceder privilégios adicionais, você não pode revogar qualquer permissão concedida por equipes de todo o site ou por projeto adicionando outra equipe personalizada.

Exemplo:

Se você quiser (por qualquer motivo) permitir a tradução para um idioma específico (digamos, *tcheco*) apenas para um conjunto fechado de tradutores confiáveis, mantendo as traduções para outros idiomas públicos, você terá que:

1. Remover a permissão para traduzir *Tcheco* de todos os usuários. Na configuração padrão, isso pode ser feito alterando a *equipe padrão* *Usuários*.

Tabela1: Grupo *Usuários*

Seleção de idioma	Como definido
Idiomas	Todos, exceto <i>Tcheco</i>

2. Adicionar uma equipe dedicada para tradutores(as) de *Tcheco*.

Tabela2: Grupo *Tradutores(as) de tcheco*

Funções	Usuários(as) avançados(as)
Seleção de projeto	<i>Todos os projetos públicos</i>
Seleção de idioma	<i>Como definido</i>
Idiomas	<i>Tcheco</i>

3. Adicionar usuários aos quais deseja conceder permissões a esta equipe.

Como você pode ver, o gerenciamento de permissões dessa maneira é poderoso, mas pode ser um trabalho tedioso. Você não pode delegá-lo a outro usuário, a menos que conceda permissões de superusuário.

Usuários, papéis, equipes e permissões

Os modelos de autenticação consistem em vários objetos:

Permissão

Permissão individual definida pelo Weblate. Permissões não podem ser atribuídas a usuários. Isso só pode ser feito através da atribuição de papéis.

Papel

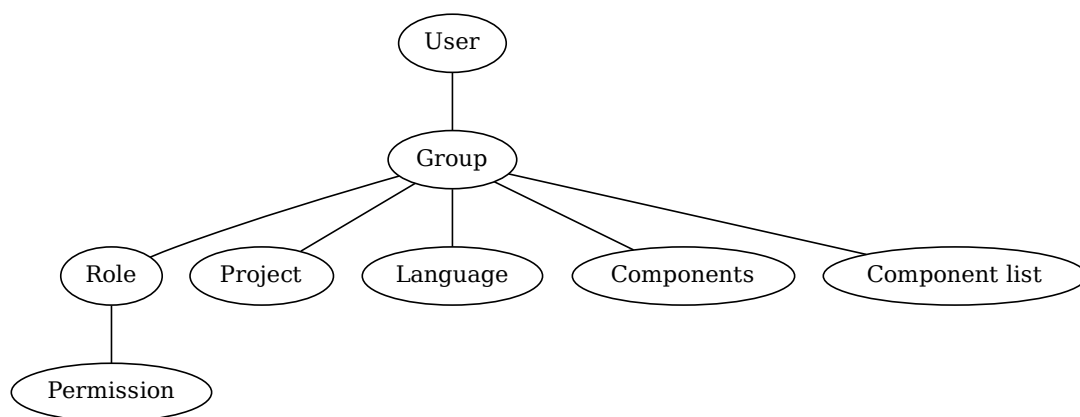
Um papel define um conjunto de permissões. Isso permite a reutilização desses conjuntos em vários lugares, facilitando a administração.

Usuário

Usuário pode pertencer a várias equipes.

Grupo

Grupo conecta papéis, usuários e objetos de autenticação (projetos, idiomas e listas de componentes).



Nota: Uma equipe não pode ter papéis atribuídos a ela; nesse caso, o acesso para navegar no projeto por qualquer pessoa é presumido (veja abaixo).

Acesso para navegar até um projeto

Um usuário tem que ser um membro de uma equipe vinculada ao projeto ou qualquer componente dentro daquele projeto. Ser membro é suficiente, não são necessárias permissões específicas para navegar o projeto (isso é usado na equipe padrão *Visualizadores*, consulte [Lista de equipes](#)).

Acesso para navegar até um componente

Um usuário pode acessar componentes irrestritos assim que for possível acessar o projeto dos componentes (e terá todas as permissões que foram concedidas ao usuário para o projeto). Com o *Acesso restrito* ativado, o acesso ao componente requer permissões explícitas para o componente (ou uma lista de componentes na qual o componente está contido).

Escopo de equipes

O escopo da permissão atribuída pelas funções nas equipes é aplicado pelas seguintes regras:

- Se a equipe especificar qualquer *Lista de componentes*, todas as permissões dadas aos membros dessa equipe serão concedidas para todos os componentes nas listas de componentes anexadas à equipe, e um acesso sem permissões adicionais é concedido para todos os projetos estes componentes estão contidos. *Componentes* e *Projetos* são ignorados.
- Se a equipe especificar quaisquer *Componentes*, todas as permissões dadas aos membros dessa equipe são concedidas para todos os componentes anexados à equipe, e um acesso sem permissões adicionais é concedido para todos os projetos nos quais que esses componentes estão. *Projetos* são ignorados.
- Caso contrário, se a equipe especificar *Projetos*, seja listando-os diretamente ou tendo *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 uma equipe são aplicadas separadamente, quando é verificado se um usuário tem acesso para realizar determinadas ações. Ou seja, é aplicado apenas a ações diretamente relacionadas ao próprio processo de tradução, como revisar, salvar 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 a seguinte equipe:

Tabela3: Grupo *Revisores-admins de espanhol*

Funções	<i>Revisar textos, Gerenciar repositório</i>
Componentes	<code>foo/bar</code>
Idiomas	<i>Espanhol</i>

Membros desta equipe terão as seguintes permissões (presumindo as configurações padrão de papéis):

- Acesso geral (navegação) para todo o projeto `foo` incluindo ambos componentes nele: `foo/bar` e `foo/baz`.
- Revisar textos na tradução para espanhol de `foo/bar` (não em outro lugar).
- Gerenciar VCS para todo o repositório `foo/bar`, por exemplo, fazer commit de alterações pendentes feitas por tradutores de todos os idiomas.

Atribuições de equipe automáticas

Na parte inferior da página de edição *Grupo* na *interface administrativa do Django*, você pode especificar *Associações automáticas de equipe*, que é uma lista de expressões regulares usadas para atribuir automaticamente usuários recém-criados a uma equipe com base em seus endereços de e-mail. Esta associação só acontece após a criação da conta.

O caso de uso mais comum para o recurso é atribuir todos os novos usuários a alguma equipe padrão. Para fazer isso, você 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 de sua empresa por padrão. Supondo que todos eles usem endereços de e-mail corporativos em seu domínio, isso pode ser feito com uma expressão como `^.*@minhaempresa.com`.

Nota: Associação automática de equipe para *Usuários* e *Visualizadores* sempre será recriada ao atualizar de uma versão anterior do Weblate para outra. Se você deseja desativá-los, defina a expressão regular para `^$` (que nunca corresponderá).

Nota: Por enquanto, não há como adicionar em massa usuários já existentes a alguma equipe por meio da interface do usuário. Para isso, você pode recorrer ao uso da [API REST](#).

Equipes e funções padrão

Após a instalação, um conjunto padrão de equipes é criado (veja [Lista de equipes](#)).

Esses funções e essas equipes são criados após a instalação. As funções embutidas são sempre mantidas atualizadas pela migração do banco de dados ao atualizar. Você não pode alterá-las. Defina uma nova função caso você queira definir seu próprio conjunto de permissões.

Lista de privilégios e funções embutidas

Escopo	Permissão	Funções
Cobrança (consulte Cobrança)	Visualizar informações de cobrança	<i>Administração, Cobrança</i>
Alterações	Baixar alterações	<i>Administração</i>
Comentários	Publicar comentário	<i>Administração, Editar fonte, Usuário avan</i>
	Excluir comentário	<i>Administração</i>
	Resolver comentário	<i>Administração, Revisar textos</i>
	Editar configurações do componente	<i>Administração</i>
Componente	Bloquear componente, evitando traduções	<i>Administração</i>
Glossário	Adicionar entrada no glossário	<i>Administração, Gerenciar glossário, Usuário avan</i>
	Editar entrada do glossário	<i>Administração, Gerenciar glossário, Usuário avan</i>
	Excluir entrada do glossário	<i>Administração, Gerenciar glossário, Usuário avan</i>
	Enviar entradas ao glossário	<i>Administração, Gerenciar glossário, Usuário avan</i>
Sugestões automáticas	Usar sugestões automáticas	<i>Administração, Editar fonte, Usuário avan</i>
Memória de tradução	Editar memória de tradução	<i>Administração, Gerenciar memória de tra</i>
	Excluir memória de tradução	<i>Administração, Gerenciar memória de tra</i>
Projetos	Editar configurações do projeto	<i>Administração</i>
	Gerenciar acesso ao projeto	<i>Administração</i>
Relatórios	Relatórios de download	<i>Administração</i>
Capturas de tela	Adicionar captura de tela	<i>Administração, Gerenciar capturas de tela</i>
	Editar captura de tela	<i>Administração, Gerenciar capturas de tela</i>
	Excluir captura de tela	<i>Administração, Gerenciar capturas de tela</i>
Textos fonte	Editar informações adicionais do texto	<i>Administração, Editar fonte</i>
Textos	Adicionar novo texto	<i>Administração</i>
	Remover um texto	<i>Administração</i>
	Dispensar verificação com falha	<i>Administração, Editar fonte, Usuário avan</i>
	Editar textos	<i>Administração, Editar fonte, Usuário avan</i>
	Revisar textos	<i>Administração, Revisar textos</i>
	Editar texto quando as sugestões são forçadas	<i>Administração, Revisar textos</i>
	Editar textos fonte	<i>Administração, Editar fonte, Usuário avan</i>
Sugestões	Aceitar sugestão	<i>Administração, Editar fonte, Usuário avan</i>
	Adicionar sugestão	<i>Administração, Editar fonte, Adicionar su</i>
	Excluir sugestão	<i>Administração, Usuário avançado</i>
	Votar na sugestão	<i>Administração, Editar fonte, Usuário avan</i>
Traduções	Adicionar idioma para tradução	<i>Administração, Usuário avançado, Geren</i>
	Executar tradução automática	<i>Administração, Tradução automática</i>
	Excluir tradução existente	<i>Administração, Gerenciar idiomas</i>
	Baixar arquivo de tradução	<i>Administração, Editar fonte, Acessar repo</i>
	Adicionar vários idiomas para tradução	<i>Administração, Gerenciar idiomas</i>
Envios	Definir autor da tradução enviada	<i>Administração</i>

Tabela 4 – continuação da página anterior

Escopo	Permissão	Funções
VCS	Sobrescrever os textos existentes com os enviados	<i>Administração, Editar fonte, Usuário avançado</i>
	Enviar traduções	<i>Administração, Editar fonte, Usuário avançado</i>
	Acessar repositório interno	<i>Administração, Acessar repositório, Usuário avançado</i>
	Fazer commit das alterações para o repositório interno	<i>Administração, Gerenciar repositório</i>
	Fazer push das alterações do repositório interno	<i>Administração, Gerenciar repositório</i>
	Redefinir as alterações no repositório interno	<i>Administração, Gerenciar repositório</i>
	Visualizar local do repositório upstream	<i>Administração, Acessar repositório, Usuário avançado</i>
	Atualizar o repositório interno	<i>Administração, Gerenciar repositório</i>
Privilégios para todo o site	Usar interface de gerenciamento	
	Adicionar novos projetos	
	Adicionar definições de idioma	
	Gerenciar definições de idiomas	
	Gerenciar equipes	
	Gerenciar usuários	
	Gerenciar funções	
	Gerenciar anúncios	
	Gerenciar memória de tradução	
	Gerenciar tradução automática	
	Gerenciar listas de componentes	

Nota: Privilégios para todo o site não são concedidos a nenhuma função padrão. Eles são poderosos e muito próximos do status de superusuário. A maioria deles afetam todos os projetos de sua instalação do Weblate.

Lista de equipes

As seguintes equipes são criadas após a instalação (ou após a execução de *setupgroups*) e você está livre para modificá-las. A migração irá, no entanto, recriá-las se você excluí-las ou renomeá-las.

Convidados

Define permissões para usuários não autenticados.

Esta equipe contém apenas usuários anônimos (consulte *ANONYMOUS_USER_NAME*).

Você pode remover funções desta equipe para limitar as permissões para usuários não autenticados.

Funções padrão: *Adicionar sugestão, Acessar repositório*

Visualizadores

Essa função garante a visibilidade de projetos públicos para todos os usuários. Por padrão, todos os usuários são membros desta equipe.

Por padrão, *associação automática de equipe* faz com que todos os novos membros de contas desta equipe quando eles entram.

Funções padrão: nenhuma

Usuários

Equipe padrão para todos os usuários.

Por padrão, *associação automática de equipe* faz com que todos os novos membros de contas desta equipe quando eles entram.

Funções padrão: *Usuário avançado*

Revisores

Grupo para revisores (consulte *Fluxos de trabalho de tradução*).

Funções padrão: *Revisar textos*

Gerenciadores

Grupo para administradores.

Funções padrão: *Administração*

Aviso: Nunca remova as equipes e usuários predefinidos do Weblate, pois isso pode levar a problemas inesperados! Se você não tiver uso para eles, você pode remover todos os privilégios deles.

2.6.3 Restrições adicionais de acesso

Se você quer usar a instalação do Weblate de forma menos pública, ou seja, permitir novos usuários apenas por convite, isso pode ser feito configurando o Weblate para que apenas usuários conhecidos tenham acesso a ele. Para fazer isso, você pode definir `REGISTRATION_OPEN` para `False` para prevenir registros de quaisquer novos usuários, e definir `REQUIRE_LOGIN` para `/*` para exigir login para acessar todas as páginas do site. Assim você terá seu Weblate privado.

Dica: Você pode usar *Novo convite de usuário* para adicionar novos usuários.

2.7 Projetos de tradução

2.7.1 Organização de tradução

O Weblate organiza o conteúdo VCS traduzível do projeto/componentes em uma 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 um aplicativo em várias versões e/ou documentação que o acompanha).
- No nível acima, *Configuração de componente*, que é na verdade o componente a ser traduzido, você define o repositório VCS a ser usado e a máscara dos arquivos a serem traduzidos.
- Acima da *Configuração de componente* existem traduções individuais, manipuladas automaticamente pelo Weblate como arquivos de tradução (que correspondem a *Máscara de arquivo* 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 arquivos suportados*.

Nota: Você pode compartilhar repositórios VCS clonados usando *URLs internas do Weblate*. Usar este recurso é altamente recomendado quando você tem muitos componentes compartilhando 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 você 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 em suas permissões, novos projetos e componentes de tradução podem ser criados. É sempre permitido para usuários com a permissão *Adicionar novos projetos*, e se sua instância usa cobrança (por exemplo, como <https://hosted.weblate.org/>, veja *Cobrança*), você também pode criar aqueles com base na permissão de seus planos da conta de usuário que gerencia a cobrança.

Você pode visualizar seu plano de cobrança atual em uma página separada:

 Weblate


Dashboard

Projects ▾

Languages ▾

Checks ▾

+ ▾

 ▾

...

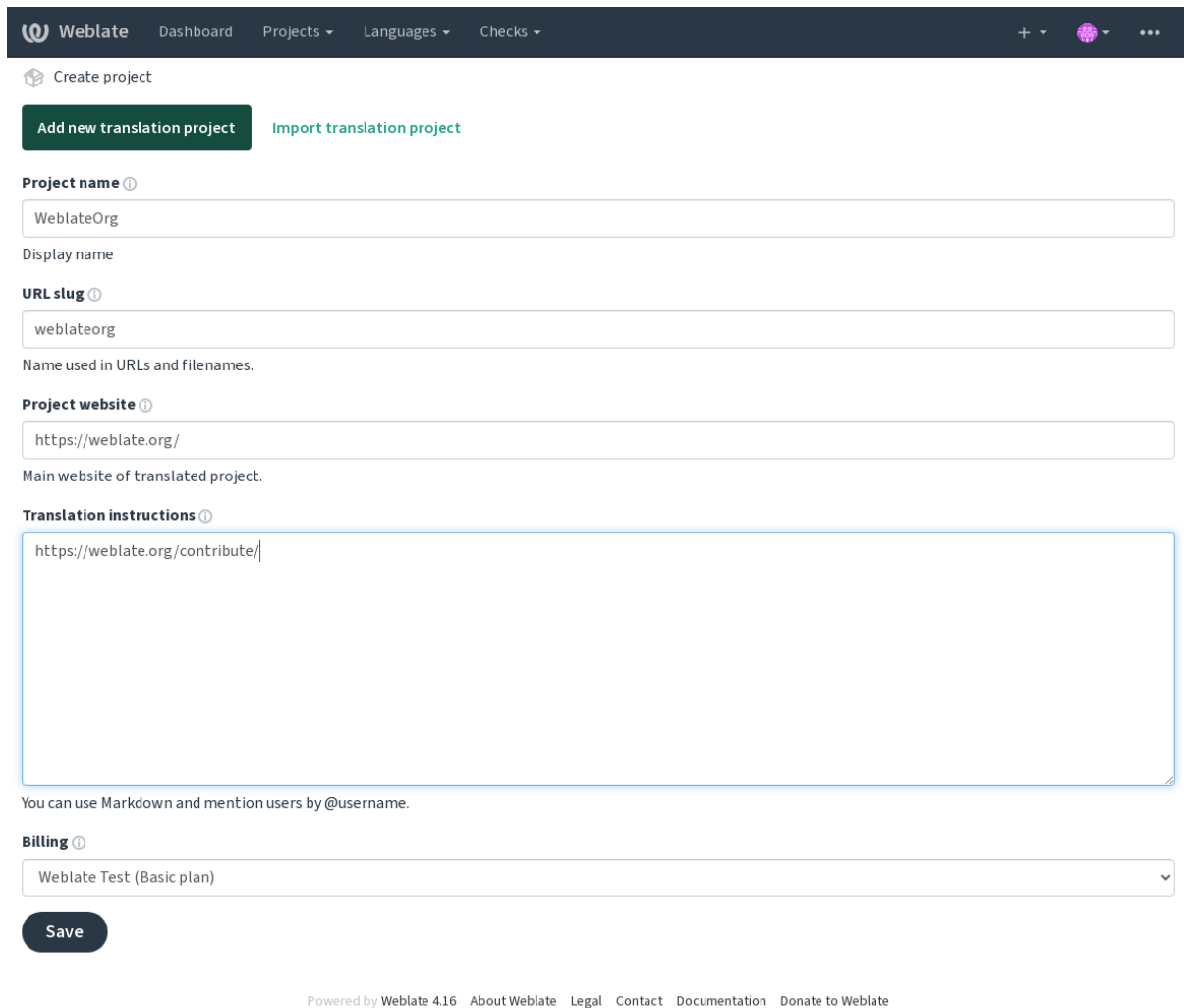
Your profile / Billing

Billing plan		ⓘ
Current plan	Basic plan (Active)	
Monthly price	19 EUR	
Yearly price	199 EUR	
Strings limit	Used 0	<div></div>
Languages limit	Used 0	<div></div>
Last invoice	2023-02-28 - 2023-03-02	
Projects limit	Used 0 of 1	<div></div>
Projects	No projects currently assigned!	
<div>Add new translation project</div>		
<div>Terminate billing plan</div>		

Invoices		
Invoice period	Invoice amount	Download invoice
02/28/2023 - 03/02/2023	19.0 EUR	Not available

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

A criação do projeto pode ser iniciada a partir daí, ou utilizando o menu da barra de navegação, preenchendo informações básicas sobre o projeto de tradução para completar a adição dele:

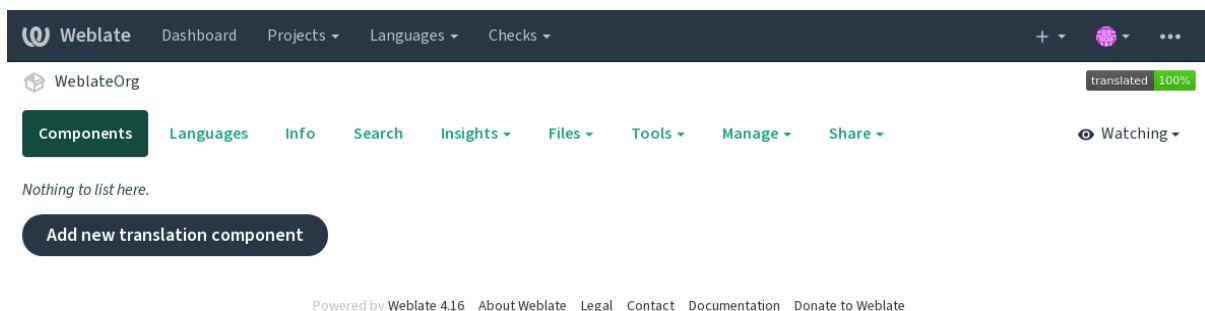


The screenshot shows the 'Create project' form in the Weblate interface. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, a 'Create project' button is visible. The form itself has two main buttons: 'Add new translation project' (highlighted in green) and 'Import translation project'. The form fields include:

- Project name**: A text input field containing 'WeblateOrg'.
- Display name**: A label for the project's display name.
- URL slug**: A text input field containing 'weblateorg'.
- Name used in URLs and filenames**: A label for the slug.
- Project website**: A text input field containing 'https://weblate.org/'.
- Main website of translated project**: A label for the website.
- Translation instructions**: A large text area containing 'https://weblate.org/contribute/'.
- You can use Markdown and mention users by @username**: A note below the instructions.
- Billing**: A dropdown menu showing 'Weblate Test (Basic plan)'.
- Save**: A dark button at the bottom of the form.

 At the very bottom, a footer line reads: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

Depois de criar o projeto, você é levado diretamente para a página do projeto:



The screenshot shows the project page for 'WeblateOrg' in the Weblate interface. The navigation bar is the same as in the previous screenshot. Below it, the project name 'WeblateOrg' is displayed, along with a 'translated 100%' badge. A secondary navigation bar contains links: 'Components' (highlighted in green), 'Languages', 'Info', 'Search', 'Insights', 'Files', 'Tools', 'Manage', and 'Share'. There is also a 'Watching' button. Below this, a message says 'Nothing to list here.' and a button 'Add new translation component' is visible. The same footer as the previous screenshot is at the bottom.

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:

De controle de versão

Cria um componente do repositório de controle de versão remoto.

De componente existente

Cria um componente adicional ao existente, escolhendo arquivos diferentes.

Ramo adicional

Cria um componente adicional ao existente, apenas para um ramo diferente.

Enviar arquivos de tradução

Envia os arquivos de tradução para o Weblate caso você não tenha controle de versão ou não queira integrá-lo ao Weblate. Você pode atualizar o conteúdo posteriormente usando a interface web ou [API REST do Weblate](#).

Traduzir documento

Envia um único documento ou arquivo de tradução e o traduz.

Iniciar do zero

Cria um projeto de tradução vazio e adiciona textos manualmente.

Depois de ter os componentes de tradução existentes, você também pode adicionar facilmente novos para arquivos adicionais ou ramos usando o mesmo repositório.


Primeiro você precisa preencher o nome e o local do repositório:

The screenshot shows the 'Create component' form in the Weblate web interface. The form is titled 'Create component' and has a dark header bar with the Weblate logo and navigation links: 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below the header, there are four tabs: 'From version control' (selected), 'Upload translations files', 'Translate document', and 'Start from scratch'. The form fields are as follows:

- Component name**: A text input field with the value 'Language names'.
- Display name**: A text input field with the value 'language-names'.
- URL slug**: A text input field with the value 'language-names'.
- Use as a glossary**: A checkbox that is unchecked.
- Project**: A dropdown menu with the value 'WeblateOrg'.
- Source language**: A dropdown menu with the value 'English'.
- Version control system**: A dropdown menu with the value 'Git'.
- Source code repository**: A text input field with the value 'https://github.com/WeblateOrg/demo.git'.
- Repository branch**: A text input field that is empty.

At the bottom of the form, there is a 'Continue' button. Below the form, there is a footer with the text 'Powered by Weblate 4.16' and links to 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Na próxima página, você verá uma lista de recursos traduzíveis descobertos:



The screenshot shows the Weblate web interface. At the top is a dark navigation bar with the Weblate logo, 'Dashboard', 'Projects', 'Languages', and 'Checks' menus. Below this is a 'Create component' button. The main content area is a modal titled 'Add new translation component'. It contains a section 'Choose translation files to import' with four radio button options, each showing a file format and a file mask. A 'Continue' button is at the bottom of the modal. At the very bottom of the page, there is a footer with the text 'Powered by Weblate 4.16' and several links: 'About Weblate', 'Legal', 'Contact', 'Documentation', and 'Donate to Weblate'.

Webate Dashboard Projects Languages Checks

Create component

Add new translation component ⓘ

Choose translation files to import ⓘ

- ☐ Specify configuration manually
- ☐ File format `Android String Resource`, File mask `app/src/main/res/values-*/strings.xml`
- ☐ File format `gettext PO file`, File mask `weblate/langdata/locale/*/LC_MESSAGES/django.po`
- ☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/django.po`
- ☐ File format `gettext PO file`, File mask `weblate/locale/*/LC_MESSAGES/djangojs.po`

Continue

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Como última etapa, você analisa as informações do componente de tradução e preenche os detalhes opcionais:

Weblate

Dashboard

Projects

Languages

Checks

Create component

Detected license as MIT, please check whether it is correct.

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.

Adding new translation

Create new language file

How to handle requests for creating new translations.

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

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 4.16

About Weblate

Legal

Contact

Documentation

Donate to Weblate

Ver também:

A interface administrativa do Django, Configuração de projeto, Configuração de componente

2.7.3 Configuração de projeto

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 seu dicionário; as traduções também são propagadas automaticamente por todos os componentes em um único projeto (a menos que desligado na configuração do componente), veja *Memória de tradução*.

Ver 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.

Ver também:

`PROJECT_NAME_RESTRICT_RE`

URL amigável

Nome do projeto adequado para URLs.

Site 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`.

Ver também:

`PROJECT_WEB_RESTRICT_RE`

Instruções de tradução

Texto que descreve o processo de localização no projeto e qualquer outra informação útil para tradutores. Markdown pode ser usado para formatação de texto ou inserção de links.

Definir o cabeçalho do “Language-Team”

Se o Weblate deve gerenciar o cabeçalho `Language-Team` (este é um recurso apenas do *GNU gettext* no momento).

Usar memória de tradução compartilhada

Para usar memória de tradução compartilhada, veja *Memória de tradução compartilhada* para mais detalhes.

O valor padrão pode ser alterado por `DEFAULT_SHARED_TM`.

Contribuir com memória de tradução compartilhada

Para contribuir com a memória de tradução compartilhada, veja *Memória de tradução compartilhada* para mais detalhes.

O valor padrão pode ser alterado por `DEFAULT_SHARED_TM`.

Controle de acesso

Configura o controle de acesso por projeto, veja *Controle de acesso do projeto* para mais detalhes.

O valor padrão pode ser alterado por `DEFAULT_ACCESS_CONTROL`.

Habilitar revisões

Habilita o fluxo de trabalho de revisão para traduções, veja *Revisores dedicados*.

Habilitar revisões de fontes

Habilita o fluxo de trabalho de revisão para textos fonte, veja *Revisões de textos fonte*.

Ver também:

report-source, *Comentários*

Habilitar ganchos

Se *Ganchos de notificação* não autenticados devem ser usados para este repositório.

Ver também:

Arquivo de idioma intermediário, *Rota de qualidade para os textos fonte*, *Formatos bilíngues e monolíngues*, *Definições de idioma*

Aliases de idioma

Define o mapeamento dos códigos de idioma ao importar traduções para o Weblate. Use isso quando os códigos de idioma forem inconsistentes em seus repositórios e você quiser obter uma visualização consistente no Weblate ou no caso de desejar usar uma nomenclatura fora do padrão de seus arquivos 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 arquivos de tradução e as correspondências diferenciam maiúsculas de minúsculas, por isso certifique-se de usar os códigos de idioma fonte na mesma forma usada nos nomes dos arquivos.

Ver também:

Adicionando novas traduções, Código do idioma, Analisando códigos de idioma

2.7.4 Configuração de componente

Um componente é um agrupamento de algo para tradução. Você insere um local de repositório VCS e uma máscara de arquivo para os arquivos que deseja traduzir, e o Weblate busca automaticamente neste VCS e encontra todos os arquivos traduzíveis correspondentes.

Ver também:

`/devel/integration`

Você pode encontrar alguns exemplos de configurações típicas nos *Formatos de arquivos suportados*.

Nota: Recomenda-se manter os componentes de tradução em um tamanho razoável – divida a tradução por qualquer coisa que faça sentido no seu caso (aplicativos ou extensões individuais, capítulos de livros ou sites).

O Weblate lida facilmente com traduções com 10000s de textos, 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 faltando, uma definição vazia é criada e nomeada como “cs_CZ (generated)”. Você 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 controle de versão

VCS para usar, veja *Integração com controle de versão* para detalhes.

Ver também:

Fazendo push das alterações do Weblate

Repositório do código-fonte

Repositório VCS usado para fazer pull das alterações.

Ver também:

Veja *Acessando repositórios* para mais detalhes sobre especificar URLs.

Dica: Pode ser uma URL VCS real ou `weblate://projeto/componente` indicando que o repositório deve ser compartilhado com outro componente. Veja *URLs internas do Weblate* para mais detalhes.

URL de push 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, isso não é usado e se aplica a definição do componente vinculado.

Ver 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 arquivos fonte (local das mensagens usadas). Quando vazio, nenhum desses links será gerado. Você 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 seus caminhos serem relativos a pastas diferentes (caminho contém `..`), você pode querer 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 *Localização contínua* não é usada, ou quando há a necessidade de mesclar manualmente as alterações. Você 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, isso não é usado e se aplica a definição do componente vinculado.

Ramo do push

Ramo para fazer push de alterações, deixe vazio para usar *Ramo do repositório*.

Para repositórios vinculados, isso não é usado e se aplica a definição do componente vinculado.

Nota: Atualmente, isso é compatível apenas com Git, GitLab e GitHub, mas é ignorado para outras integrações VCS.

Ver também:

Fazendo push das alterações do Weblate

Máscara de arquivo

Máscara de arquivos a traduzir, incluindo caminho. Deve incluir um “*” substituindo o código de idioma (veja *Definições de idioma* para informações sobre como isso é processado). Caso seu repositório contenha mais de um arquivo de tradução (por exemplo, mais domínios gettext), você precisa criar um componente para cada um deles.

Por exemplo, `po/* .po` ou `locale/*/LC_MESSAGES/django .po`.

No caso de seu nome de arquivo conter caracteres especiais como `[`, `]`, eles precisam ser escapados como `[]` ou `[]`.

Ver também:

Formatos bilíngues e monolíngues, O que significa “Existem mais arquivos para o idioma único (en)”?

Arquivo de idioma da base monolíngue

Arquivo base contendo definições de texto para *Componentes monolíngues*.

Ver também:

Formatos bilíngues e monolíngues, O que significa “Existem mais arquivos para o idioma único (en)”?

Editar o arquivo base

Permite a edição do arquivo base para *Componentes monolíngues*.

Arquivo de idioma intermediário

Arquivo de idioma intermediário para *Componentes monolíngues*. Na maioria dos casos, este é um arquivo de tradução fornecido por desenvolvedores e é usado ao criar textos fonte reais.

Quando definido, os textos fonte são baseados neste arquivo, mas todos os outros idiomas são baseados no *Arquivo de idioma da base monolíngue*. Caso o texto não esteja traduzido para o idioma fonte, a tradução para outros idiomas é proibida. Isso fornece uma *Rota de qualidade para os textos fonte*.

Ver também:

Rota de qualidade para os textos fonte, Formatos bilíngues e monolíngues, O que significa “Existem mais arquivos para o idioma único (en)”?

Modelo para novas traduções

Arquivo base usado para gerar novas traduções, por exemplo, arquivo `.pot` com `gettext`.

Dica: Em muitos formatos monolíngues, o Weblate começa com um arquivo vazio por padrão. Use-o no caso de desejar que todos os textos estejam presentes com um valor vazio ao criar uma nova tradução.

Ver também:

[adding-translation](#), *Adicionando novas traduções*, *Adicionando nova tradução*, *Formatos bilíngues e monolíngues*, *O que significa “Existem mais arquivos para o idioma único (en)”?*

Formato de arquivo

Formato do arquivo de tradução, veja também *Formatos de arquivos suportados*.

Endereço do relatório de erros do texto fonte

Endereço de e-mail usado para relatar bugs upstream. Este endereço também receberá notificação sobre quaisquer comentários de texto fonte feitos no Weblate.

Permitir propagação de tradução

Você pode desativar a propagação de traduções para este componente de outros componentes dentro do mesmo projeto. Isso realmente depende do que você está traduzindo, 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 você esteja usando os mesmos IDs em todo o projeto.

O valor padrão pode ser alterado por `DEFAULT_TRANSLATION_PROPAGATION`.

Ver também:

Mantendo traduções iguais entre componentes

Habilitar sugestões

Se sugestões de tradução são aceitas para este componente.

Votação de sugestões

Ativa a emissão de votos para sugestões, veja *Votação de sugestões*.

Aceitar sugestões automaticamente

Aceita automaticamente sugestões votadas, veja *Votação de sugestões*.

Marcadores de tradução

Personalização de verificações de qualidade e outros comportamentos do Weblate, veja *Personalizando comportamento usando marcadores*.

Verificações forçadas

Lista de verificações que não podem ser ignoradas, veja *Forçando verificações*.

Nota: Forçar a verificação não a habilita automaticamente, você ainda deve habilitá-la *Personalizando comportamento usando marcadores* em *Marcadores de tradução* ou *Informações adicionais sobre textos fonte*.

Licença da tradução

Licença da tradução (não precisa ser igual à licença do código-fonte).

Acordo de colaborador

Acordo de usuário que precisa ser aprovado antes de um usuário poder traduzir este componente.

Adicionando nova tradução

Como lidar com solicitações de criação de novos idiomas. Opções disponíveis:

Contatar os mantenedores

O usuário pode selecionar o idioma desejado e os mantenedores do projeto receberão uma notificação sobre isso. Cabe a eles adicionar (ou não) o idioma ao repositório.

Apontar para a URL de instruções de tradução

É apresentado ao usuário um link para a página que descreve o processo de iniciar novas traduções. Use-o caso deseje um processo mais formal (por exemplo, formar uma equipe antes de iniciar a tradução real).

Criar novo arquivo de idioma

O usuário pode selecionar o idioma e o Weblate cria automaticamente o arquivo para ele e a tradução pode começar.

Desabilitar adição de novas traduções

Não haverá a opção para o usuário iniciar uma nova tradução.

Dica: Os administradores do projeto podem adicionar novas traduções mesmo se estiver desabilitado aqui quando for possível (tanto *Modelo para novas traduções* ou o formato de arquivo tem suporte a partir de um arquivo vazio).

Ver também:

adding-translation, *Adicionando novas traduções*

Gerenciar textos

Novo na versão 4.5.

Configura se os usuários 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 novos textos no Weblate deve ser desabilitado (elas seriam descartadas na próxima vez que você atualizasse os arquivos de tradução). No Weblate, você pode gerenciar textos para cada tradução e não exige que os textos em todas as traduções sejam consistentes.

Para formatos monolíngues, os textos são gerenciados apenas no idioma fonte e são automaticamente adicionados ou removidos nas traduções. Os textos aparecem nos arquivos de tradução assim que são traduzidos.

Ver também:

Formatos bilíngues e monolíngues, *adding-new-strings*, `POST /api/translations/(string:project)/(string:component)/(string:language)/units/`

Estilo de código de idioma

Personalize o código de idioma usado para gerar o nome do arquivo para traduções criado por Weblate.

Ver também:

Adicionando novas traduções, *Código do idioma*, *Analizando códigos de idioma*

Estilo de mesclagem

Você pode configurar como as atualizações do repositório upstream são tratadas. Isso pode não ser compatível com alguns VCSs. A implementação real depende do VCS, veja /vcs.

Rebase

Faz um rebase dos commits do Weblate no topo do repositório upstream na atualização. Isso fornece um histórico limpo sem commits de mesclagem extras.

Rebasing pode causar problemas em caso de mesclagens complicadas, então considere cuidadosamente se você quer ou não habilitá-los.

Você pode precisar habilitar o push forçado escolhendo *Git com push forçado* como *Sistema de controle de versão*, especialmente ao fazer push para um branch diferente.

Mesclar

Alterações no repositório upstream são mesclados com o do Weblate. Esta configuração utiliza fast-forward quando possível. Esta é a maneira mais segura, mas pode produzir muitos commits de mesclagem.

Mesclar sem fast-forward

As alterações do repositório upstream são mescladas com o do Weblate fazendo uma mesclagem de commits toda vez (mesmo quando fast-forward era possível). Toda alteração no Weblate vai aparecer como um commit de merge no repositório do Weblate.

Valor padrão pode ser alterado por `DEFAULT_MERGE_STYLE`.

Mensagens de commit, add, merge, extensão e merge request

Mensagem usada ao fazer commit uma tradução, veja *Marcação de modelo*.

O valor padrão pode ser alterado por `DEFAULT_ADD_MESSAGE`, `DEFAULT_ADDON_MESSAGE`, `DEFAULT_COMMIT_MESSAGE`, `DEFAULT_DELETE_MESSAGE`, `DEFAULT_MERGE_MESSAGE`, `DEFAULT_PULL_MESSAGE`.

Push ao fazer commit

Se as alterações confirmadas devem ser enviadas automaticamente para o repositório upstream. Quando habilitado, o push é iniciado assim que o Weblate confirma as alterações em seu repositório subjacente (veja *Commits adiados*). Para habilitar o push *URL de push do repositório* também deve ser configurado.

Idade das alterações para fazer commit

Define a idade (em horas) das alterações antes de ser feito commit pela tarefa em segundo plano ou pelo comando de gerenciamento `commit_pending`. Todas as alterações em um componente tem commit feito quando houver pelo menos uma alteração anterior a esse período.

O valor padrão pode ser alterado por `COMMIT_PENDING_HOURS`.

Dica: Existem outras situações onde mudanças pendentes podem ter seu commit feito, veja *Commits adiados*.

Bloquear em erro

Bloqueia o componente (e os componentes vinculados, consulte *URLs internas do Weblate*) no primeiro push ou mesclagem com falha em seu repositório upstream, 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 você estiver traduzindo de algo diferente do inglês.

Dica: Caso você esteja traduzindo arquivos bilíngues do inglês, mas também queira fazer correções na tradução para o inglês, escolha *Inglês (Desenvolvedor)* como idioma fonte para evitar conflito entre o nome do idioma fonte e a tradução existente.

Para traduções monolíngues, você pode usar tradução intermediária neste caso, veja *Arquivo de idioma intermediário*.

Filtro de idioma

Expressão regular usada para filtrar a tradução ao examinar por uma máscara de arquivo. Pode ser usada para limitar a lista de idiomas gerenciados pelo Weblate.

Nota: Você precisa listar os códigos de idioma conforme aparecem no nome do arquivo.

Alguns exemplos de filtragem:

Descrição do filtro	Expressão regular
Apenas idiomas selecionados	<code>^(cs de es)\$</code>
Exclui idiomas	<code>^(?! (it fr)\$) .+\$</code>
Filtra apenas códigos de duas letras	<code>^[.]+\$</code>
Exclui arquivos que não são idiomas	<code>^(?! (blank)\$) .+\$</code>
Inclui todos os arquivos (padrão)	<code>^[^.] +\$</code>

Expressão regular de variantes

Expressão regular usada para determinar as variantes de um texto, veja variants.

Nota: A maioria dos campos pode ser editada por proprietários ou administradores de projetos, na interface do Weblate.

Ver também:

O Weblate tem suporte a outros VCSs que não o Git e o Mercurial?, alerts

Prioridade

Componentes com prioridade mais alta são oferecidos primeiro para os tradutores.

Alterado na versão 4.15: Isso agora também afeta a ordem dos termos correspondentes do glossário.

Acesso restrito

Por padrão, o componente é visível para qualquer pessoa que tenha 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 em um componente ou nível de lista de componentes assume a permissão de acesso a um componente, independentemente das permissões de nível de projeto. Você terá que conceder acesso a ele explicitamente. Isso pode ser feito concedendo acesso a um novo grupo de usuários e colocando usuários nele, ou usando os grupos de controle de acesso padrão *personalizado* ou *privado*.

O valor padrão 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

Você pode escolher projetos adicionais onde o componente ficará visível. Útil para bibliotecas compartilhadas que você usa em vários projetos.

Nota: Compartilhar um componente não altera seu controle de acesso. Só o torna visível ao navegar em outros projetos. Os usuários 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. Você pode configurar como ele será listado usando a *Cor do glossário*.

O glossário estará acessível em todos os projetos definidos por *Compartilhar em projetos*.

É recomendado habilitar *Gerenciar textos* em glossários para permitir adicionar novas palavras a eles.

Ver 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 componente*
 - *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 arquivo 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 }}
    Porcentagem de textos que precisam de revisão

{{ stats.translated }}
    Contagem de textos traduzidos

{{ stats.translated_percent }}
    Porcentagem de textos traduzidos

{{ stats.allchecks }}
    Número de textos com falhas nas verificações

{{ stats.allchecks_percent }}
    Porcentagem 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 arquivo

{{filename}}
    nome do arquivo, você também pode remover as partes principais usando o filtro parentdir, por exemplo
    {{filename|parentdir}}
```

Você pode combiná-los com filtros:

```
{{ component|title }}
```

Você pode usar as condições:

```
{% if stats.translated_percent > 80 %}Well translated!{% endif %}
```

Há uma tag adicional disponível para substituir caracteres:

```
{% replace component "-" " " %}
```

Você pode combiná-lo com filtros:

```
{% replace component|capfirst "-" " " %}
```

Existem também filtros adicionais para manipular com nomes de arquivos:

```
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 para o Weblate pode ser um processo demorado, dependendo do tamanho de suas traduções. Aqui estão algumas dicas:

Otimizar a configuração

A configuração padrão é útil para testar e depurar o Weblate, enquanto para uma configuração de produção, você 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 usando Celery](#))
- *Habilitar o cache*
- *Usar um poderoso mecanismo de banco de dados*
- *Desabilitar o modo de depuração*

Verificar limites de recursos

Se você estiver importando grandes traduções ou repositórios, pode ser atingido por limitações de recursos do seu servidor.

- Verifique a quantidade de memória livre, pois os arquivos 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 usando Celery](#)).

Desabilitar 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 arquivos de tradução (por exemplo, para diferentes domínios gettext ou partes de aplicativos Android), você pode querer importá-los automaticamente. Isso pode ser feito na linha de comando usando `import_project` ou `import_json`, ou instalando a extensão [Descoberta de componente](#).

Para usar a extensão, primeiro você precisa criar um componente para um arquivo 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 gerenciamento, você precisa criar um projeto que conterá todos os componentes e então executar `import_project` ou `import_json`.

Ver também:

[Comandos de gerência](#), [Descoberta de componente](#)

2.8 Definições de idioma

Para apresentar diferentes traduções corretamente, são necessárias informações sobre nome do idioma, direção de texto, definições plurais e código de idioma.

2.8.1 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 desabilitado usando `UPDATE_LANGUAGES`. Você também pode forçar a atualização do banco de dados para corresponder aos dados embutidos do Weblate usando `setuplang`.

Ver também:

[Estendendo definições embutidas de idioma](#), [Definições atuais de idiomas](#)

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 [Máscara de arquivo](#) para qualquer objeto de idioma existente.

Você pode ajustar ainda mais esse mapeamento no nível do projeto por [Aliases de idioma](#).

Se nenhuma correspondência exata for encontrada, uma tentativa será feita para melhor encaixá-la em um idioma existente. São testadas etapas a seguir:

- Procuras sem diferenciar maiúsculo de minúsculo.
- Normalizar sublinhados e traços.
- 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 usando 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)`. Você pode querer mudar isso na interface administrativa mais tarde, (veja [Alterando as definições de idioma](#)) e relatar ao rastreador de problemas (veja [Contribuindo para o Weblate](#)), para que a definição apropriada possa ser adicionada ao próximo lançamento do Weblate.

Dica: No caso de você ver algo indesejado como idioma, você pode querer ajustar [Filtro de idioma](#) para ignorar tal arquivo ao analisar traduções.

Ver também:

[Código do idioma](#), [Adicionando novas traduções](#)

2.8.3 Alterando as definições de idioma

Você 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 macroidiomas

Em muitos casos, não é uma boa ideia usar código de idioma macro para uma tradução. O caso típico problemático pode ser a língua curda, que pode ser escrita em árabe ou latim, dependendo da variante real. Para obter o comportamento correto no Weblate, é recomendável usar apenas códigos de idioma individuais e evitar macroidiomas.

Ver 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](#).

Ver também:

[Analisando códigos de idioma](#), [Adicionando novas traduções](#)

Nome do idioma

Nome visível do idioma. Os nomes dos idiomas incluídos no Weblate também estão sendo localizados dependendo do idioma da interface do usuário.

Direção do texto

Determina se o idioma é escrito da direita para a esquerda ou da esquerda para a direita. Esta propriedade é autodelectada corretamente para a maioria dos idiomas.

Número de plurais

Número 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.

Ver também:

Plurais, GNU gettext utilities: Plural forms, Language Plural Rules by the Unicode Consortium

Número de apresentadores

Número de falantes deste idioma em todo o mundo.

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 arquivo.

Weblate pode iniciar automaticamente uma nova tradução para todos os formatos de arquivo.

Alguns formatos esperam começar com um arquivo vazio e apenas textos traduzidos para serem incluídas (por exemplo, *Recurso de textos 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 necessitando edição. Em algumas situações, isso realmente não depende do formato, mas sim do framework que você usa para lidar com a tradução (por exemplo, com *Arquivos JSON*).

Quando você especifica *Modelo para novas traduções* em *Configuração de componente*, Weblate usará este arquivo para iniciar novas traduções. Todas as traduções existentes serão removidas do arquivo ao fazer isso.

Quando *Modelo para novas traduções* está vazio e o formato de arquivo suporta, um arquivo vazio é criado onde novos textos serão adicionados assim que forem traduzidos.

O *Estilo de código de idioma* permite que você personalize o código de idioma usado nos nomes de arquivos gerados:

Padrão baseado no formato de arquivo

Dependendo do formato do arquivo, para a maioria deles utiliza-se POSIX.

Estilo POSIX usando sublinhado como um separador

Normalmente usado por gettext e ferramentas relacionadas, produz códigos de linguagem como `pt_BR`.

Estilo POSIX usando sublinhado como separador, incluindo código de país

Código de idioma do estilo POSIX incluindo o código do país mesmo quando não necessário (por exemplo, `cs_CZ`).

Estilo BCP usando hífen como um separador

Normalmente usado em plataformas web, produz códigos de idioma como `pt-BR`.

Estilo BCP usando hífen como separador, incluindo código de país

Código de idioma do estilo BCP, incluindo o código do país, mesmo quando não necessário (por exemplo, `cs-CZ`).

Estilo BCP usando hífen como separador, antigos códigos de idiomas

Usa códigos legados para notação de estilo chinês e BCP.

Estilo BCP usando hífen como um separador, caixa baixa

notação estilo BCP, toda em minúsculas (por exemplo `cs-cz`).

Estilo de metadados da Apple App Store

Estilo próprio para enviar metadados para Apple App Store.

Estilo de metadados do Google Play

Estilo próprio para enviar metadados para Google Play Store.

Estilo Android

Usado apenas em aplicativos 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 de idioma* são aplicados ao contrário.

Nota: O Weblate reconhece qualquer um desses ao analisar arquivos de tradução; as configurações acima apenas influenciam como os novos arquivos são criados.

Ver também:

Código do idioma, Aliases de idioma, Analisando códigos de idioma

2.9 Localização contínua

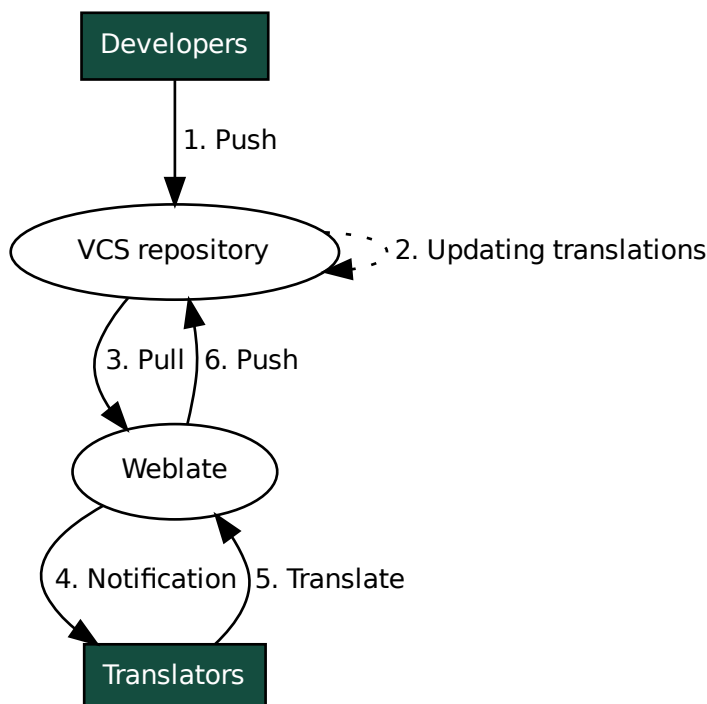
Há infraestrutura em vigor para que sua tradução acompanhe de perto o desenvolvimento. Desta forma, os tradutores podem trabalhar em traduções o tempo todo, em vez de trabalhar através de uma enorme quantidade de texto novo pouco antes do lançamento.

Ver também:

/devell/integration descreve maneiras básicas de integrar seu desenvolvimento com o Weblate.

Este é o processo:

1. Os desenvolvedores fazem alterações e fazem *push* delas para o repositório VCS.
2. Opcionalmente, os arquivos de tradução são atualizados (isso depende do formato do arquivo, consulte *Por que é que o Weblate ainda mostra os textos de tradução antigos quando eu atualizei o modelo?*).
3. O Weblate faz o *pull* das alterações do repositório VCS, consulte *Atualizando repositórios*.
4. Uma vez que o Weblate detecta alterações nas traduções, os tradutores são notificados com base em suas configurações de assinatura.
5. Os tradutores enviam traduções usando a interface web do Weblate ou enviam alterações feitas offline.
6. Uma vez que os tradutores tenham finalizado, o Weblate faz commit das alterações no repositório local (veja *Commits adiados*) e faz *push* delas de volta se tiver permissões para fazê-lo (veja *Fazendo push das alterações do Weblate*).



2.9.1 Atualizando repositórios

Você deve configurar alguma maneira de atualizar repositórios de backend a partir de sua fonte.

- Use *Ganchos de notificação* para integrar com a maioria dos serviços comuns de hospedagem de código:
 - *Recebendo automaticamente alterações do GitHub*
 - *Recebendo automaticamente alterações do GitLab*
 - *Recebendo automaticamente alterações do Bitbucket*
 - *Recebendo automaticamente alterações do Pagure*
 - *Recebendo automaticamente alterações do Azure Repos*
 - *Recebendo automaticamente alterações do Gitea Repos*
- Acione manualmente a atualização no gerenciamento do repositório ou usando *API REST do Weblate* ou *Weblate Client*
- Habilite `AUTO_UPDATE` para atualizar automaticamente todos os componentes na sua instância do Weblate
- Execute `updategit` (com a seleção de um projeto ou `--all` para atualizar tudo)

Sempre que o Weblate atualizar o repositório, as extensões de pós-atualização serão acionadas, consulte *Extensões*.

Evitando conflitos de mesclagem

Os conflitos de mesclagem do Weblate surgem quando o mesmo arquivo foi alterado tanto no Weblate quanto fora dele. Existem duas abordagens para lidar com isso - evitar edições fora do Weblate ou integrar o Weblate no seu processo de atualização, de modo que ele descarte alterações antes de atualizar os arquivos fora do Weblate.

A primeira abordagem é fácil com arquivos monolíngues - você pode adicionar novos textos no Weblate e deixar a edição completa dos arquivos lá. Para arquivos bilíngues, geralmente há algum tipo de processo de extração de mensagens para gerar arquivos traduzíveis a partir do código fonte. Em alguns casos, isso pode ser dividido em duas partes - uma para a extração gera modelo (por exemplo, o GETTEXT POT é gerado usando **xgettext**) e, em seguida, o processo mais mescla-o em traduções reais (os arquivos GETTEXT PO são atualizados usando **msgmerge**). Você pode executar o segundo passo dentro do Weblate e ele garantirá que todas as alterações pendentes sejam incluídas antes desta operação.

A segunda abordagem pode ser alcançada usando *API REST do Weblate* para forçar o Weblate a fazer push de todas as alterações pendentes e bloquear a tradução enquanto você está fazendo alterações do seu lado.

O script para fazer atualizações pode ser assim:

```
# Lock Weblate translation
wlc lock
# Push changes from Weblate to upstream repository
wlc push
# Pull changes from upstream repository to your local copy
git pull
# Update translation files, this example is for Django
./manage.py makemessages --keep-pot -a
git commit -m 'Locale updates' -- locale
# Push changes to upstream repository
git push
# Tell Weblate to pull changes (not needed if Weblate follows your repo
# automatically)
wlc pull
# Unlock translations
wlc unlock
```

Se você tiver vários componentes compartilhando o mesmo repositório, você precisa bloqueá-los todos separadamente:

```
wlc lock foo/bar
wlc lock foo/baz
wlc lock foo/baj
```

Nota: O exemplo usa *Weblate Client*, que precisa de configuração (chaves de API) para ser capaz de controlar o Weblate remotamente. Você também pode conseguir isso usando qualquer cliente HTTP em vez de wlc, por exemplo, curl, ver *API REST do Weblate*.

Ver também:

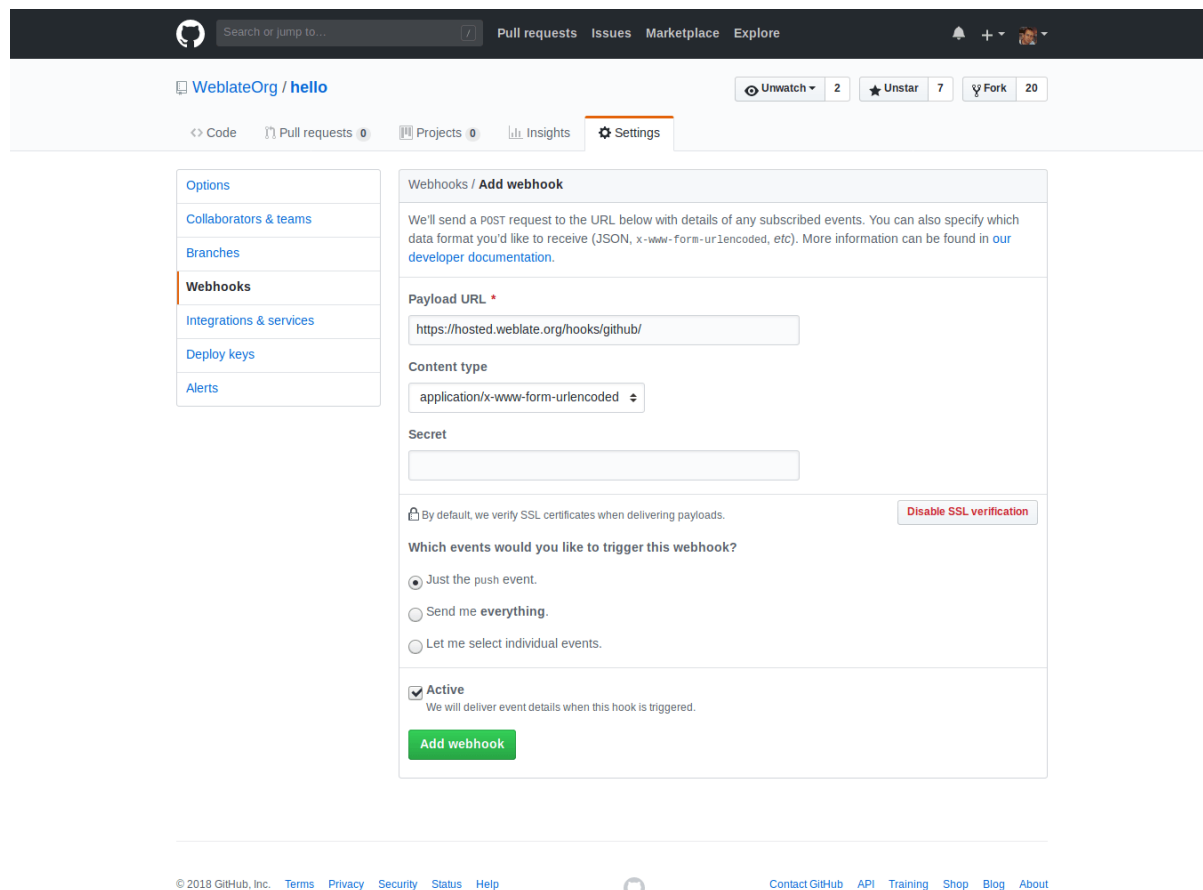
Weblate Client

Recebendo automaticamente alterações do GitHub

O Weblate vem com suporte nativo ao GitHub.

Se você estiver usando o Hosted Weblate, a abordagem recomendada é instalar o [aplicativo Weblate](#), dessa forma você terá a configuração correta sem ter que configurar muita coisa. Também pode ser usado para fazer push de mudanças de volta.

Para receber notificações em cada push a um repositório do GitHub, adicione o webhook do Weblate nas configurações do repositório (*Webhooks*) como mostrado na imagem abaixo:



Para a URL de carga útil, anexar `/hooks/github/` à URL do Weblate, por exemplo, para o serviço Hosted Weblate, este é `https://hosted.weblate.org/hooks/github/`.

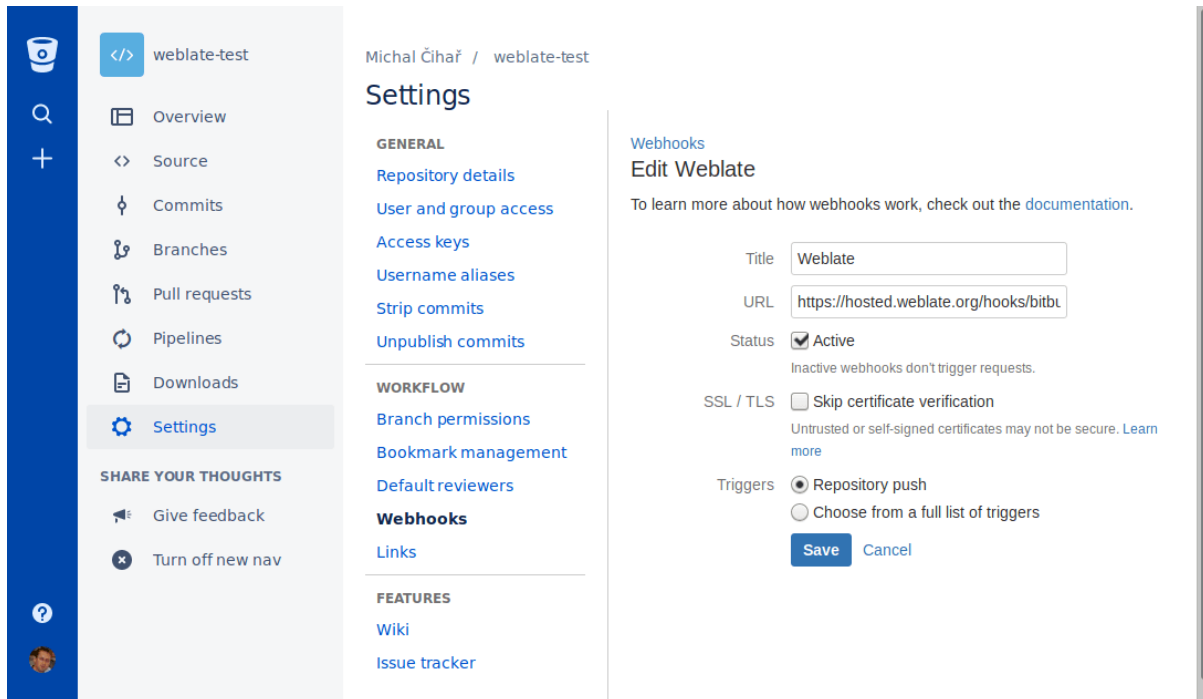
Você pode deixar outros valores nas configurações padrão (o Weblate pode lidar com ambos os tipos de conteúdo e consome apenas o evento *push*).

Ver também:

POST /hooks/github/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do Bitbucket

O Weblate tem suporte para webhooks do Bitbucket, adicione um webhook que aciona no push do repositório, com destino a URL `/hooks/bitbucket/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/bitbucket/`).



Ver também:

POST /hooks/bitbucket/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do GitLab

O Weblate tem suporte para ganchos do GitLab, adiciona um webhook de projeto com destino a URL `/hooks/gitlab/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitlab/`).

Ver também:

POST /hooks/gitlab/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do Pagine

Novo na versão 3.3.

O Weblate tem suporte para ganchos Pagine. Adicione um webhook com destino a URL `/hooks/pagine/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/pagine/`). Isso pode ser feito em *Web-hooks* em *Project options*:

The screenshot shows the Weblate interface for a project named 'nijel-test'. The top navigation bar includes the 'fedora PAGURE' logo, 'Browse', 'Create', and a user profile icon. Below the navigation bar, there are tabs for 'Source', 'Issues' (0), 'Pull Requests' (0), 'Stats', and 'Settings' (active). The 'Settings' tab is expanded, showing a sidebar with various settings categories: Project Settings, Project Details, Default Branch, Private Web Hook Key, API Keys, Project Options (selected), Public Notifications, Users & Groups, Deploy Keys, Hooks, Priorities, Roadmap, Close Status, Custom Issue Fields, Reports, Tags, Quick Replies, Regenerate Repos, Give Project, and Delete Project. The main content area is titled 'Project Options' and contains several checkboxes for enabling or disabling features. The 'Activate Web-hooks' section shows a text input field with the URL 'https://hosted.weblate.org/hooks/pagure/' and buttons for 'Update' and 'Test web-hook'. Below this, there is a 'Learn more about' section with a list of links: Flags, Tracker read-only, Pull-request access only, Roadmap on Issue page, and fedmsg notifications.

Ver também:

POST /hooks/pagure/, Acessando repositórios do Hosted Weblate

Recebendo automaticamente alterações do Azure Repos

Novo na versão 3.8.

O Weblate tem suporte para webhooks do Azure Repos, adicione um webhook para evento *Code pushed* com destino para URL `/ganchos/azure/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/azure/`). Isso pode ser feito em *Service hooks* em *Project settings*.

Ver também:

Webhooks no manual do Azure DevOps, *POST /hooks/azure/, Acessando repositórios do Hosted Weblate*

Recebendo automaticamente alterações do Gitea Repos

Novo na versão 3.9.

Weblate tem suporte para webhooks do Gitea, adicione um *Gitea Webhook* para evento *Push events* com destino para a URL `/hooks/gitea/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitea/`). Isso pode ser feito no *Webhooks* em *Settings* do repositório.

Ver também:

Webhooks no manual do Gitea, *POST /hooks/gitea/*, *Acessando repositórios do Hosted Weblate*

Recebendo automaticamente alterações de Gitee Repos

Novo na versão 3.9.

O Weblate tem suporte para webhooks Gitee, adicione um *WebHook* para o evento *Push* com destino para URL `/hooks/gitee/` na instalação do Weblate (por exemplo, `https://hosted.weblate.org/hooks/gitee/`). Isso pode ser feito em *WebHooks* sob *Management* do repositório.

Ver também:

Webhooks no manual do Gitee, *POST /hooks/gitee/*, *Acessando repositórios do Hosted Weblate*

Atualizando automaticamente repositórios *nightly*

O Weblate busca automaticamente repositórios remotos *nightly* para melhorar o desempenho ao mesclar alterações mais tarde. Você pode opcionalmente transformar isso em fazer mesclagens noturnas também, ativando *AUTO_UPDATE*.

2.9.2 Fazendo push das alterações do Weblate

Cada componente de tradução pode ter uma URL de push configurada (veja *URL de push do repositório*) e, nesse caso, o Weblate será capaz de fazer push de alteração para o repositório remoto. O Weblate também pode ser configurado para fazer push automaticamente das alterações em cada commit (isso é o padrão, veja *Push ao fazer commit*). Se você não quiser que seja feito push automático das alterações, você pode fazer isso manualmente em *Manutenção do repositório* ou usando API via *wlc push*.

As opções de push diferem com base no *Integração com controle de versão* usado, mais detalhes são encontrados nesse capítulo.

No caso de você não querer pushes diretos pelo Weblate, há suporte para pull requests do *Pull requests do GitHub*, do *Merge requests do GitLab*, do *Pull requests do Gitea* e do *Merge requests do Pagure* ou revisões do *Gerrit*, você pode ativá-los escolhendo *GitHub*, *GitLab*, *Gitea*, *Gerrit* ou *Pagure*, como *Sistema de controle de versão* em *Configuração de componente*.

No geral, as opções a seguir estão disponíveis com Git, GitHub e GitLab:

Configuração desejada	Sistema de controle de versão	URL de push do repositório	Ramo do push
Sem push	<i>Git</i>	<i>vazio</i>	<i>vazio</i>
Push diretamente	<i>Git</i>	URL de SSH	<i>vazio</i>
Fazer push para ramo separado	<i>Git</i>	URL de SSH	Nome do ramo
Pull request de GitHub do fork	<i>Pull requests do GitHub</i>	<i>vazio</i>	<i>vazio</i>
Pull request de GitHub do ramo	<i>Pull requests do GitHub</i>	URL de SSH ¹	Nome do ramo
Merge request de GitLab do fork	<i>Merge requests do GitLab</i>	<i>vazio</i>	<i>vazio</i>
Merge request de GitLab do ramo	<i>Merge requests do GitLab</i>	URL de SSH ^{Página 294, 1}	Nome do ramo
Merge request de Gitea do fork	<i>Pull requests do Gitea</i>	<i>vazio</i>	<i>vazio</i>
Merge request de Gitea do ramo	<i>Pull requests do Gitea</i>	URL de SSH ¹	Nome do ramo
Merge request de Pagure do fork	<i>Merge requests do Pagure</i>	<i>vazio</i>	<i>vazio</i>
Merge request de Pagure do ramo	<i>Merge requests do Pagure</i>	URL de SSH ¹	Nome do ramo

Nota: Você também pode habilitar o push automático de alterações após o Weblate fazer commit, isso pode ser feito em *Push ao fazer commit*.

Ver também:

Consulte *Acessando repositórios* para configurar chaves SSH e *Commits adiados* para obter informações sobre quando o Weblate decide fazer commit de alterações.

Ramos protegidos

Se você estiver usando o Weblate em ramo protegido, você pode configurá-lo para usar pull requests e executar revisão real sobre as traduções (o que pode ser problemático para idiomas que você não conhece). Uma abordagem alternativa é abrir mão desta limitação em favor do usuário de push no Weblate.

Por exemplo, no GitHub, isso pode ser feito na configuração do repositório:

¹ Pode estar vazia caso o *Repositório do código-fonte* tenha suporte a push.

☒ **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

Required approving reviews: **1** ▼

☐ **Dismiss stale pull request approvals when new commits are pushed**

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

☐ **Require review from Code Owners**

Require an approved review in pull requests including files with a designated code owner.

☒ **Restrict who can dismiss pull request reviews**

Specify people or teams allowed to dismiss pull request reviews.

🔍 Search for people or teams

People and teams that can dismiss reviews.



Organization and repository administrators

These members can always dismiss.



weblate

Weblate push user



2.9.3 Interagindo com os outros

O Weblate facilita a interação com outras pessoas usando sua API.

Ver também:

API REST do Weblate

2.9.4 Commits adiados

O comportamento do Weblate é agrupar commits do mesmo autor em um só commit, se possível. Isso reduz consideravelmente o número de commits, no entanto, você pode precisar dizer explicitamente para ele fazer os commits no caso de você querer deixar o repositório VCS em sincronia, por exemplo, para mesclagem (isso é por padrão permitido para o grupo *Gerenciadores*, consulte *Lista de privilégios e funções embutidas*).

As alterações neste modo têm seu commit feito assim que qualquer uma das seguintes condições são cumpridas:

- Outra pessoa altera um texto já alterado.
- Um merge do upstream é feito.
- Um commit explícito é solicitado.
- É solicitado o download de um arquivo.
- A alteração é mais antiga do que o período definido como *Idade das alterações para fazer commit* em *Configuração de componente*.

Dica: Os commits são criados para cada componente. Então, caso você tenha muitos componentes, você ainda verá muitos compromissos. Você pode utilizar a extensão *Squash de commits git* neste caso.

If you want to commit changes more frequently and without checking of age, you can schedule a regular task to perform a commit. This can be done using *Periodic Tasks* in *A interface administrativa do Django*. First create desired *Interval* (for example 120 seconds). Then add new periodic task and choose `weblate.trans.tasks.commit_pending` as *Task* with `{"hours": 0}` as *Keyword Arguments* and desired interval.

2.9.5 Processando repositório com scripts

A maneira de personalizar como o Weblate interage com o repositório é com *Extensões*. Consulte *Escrevendo scripts para extensões* para obter informações sobre como executar scripts externos através de extensões.

2.9.6 Mantendo traduções iguais entre componentes

Uma vez que você tenha vários componentes de tradução, você pode querer garantir que os mesmos textos tenham a mesma tradução. Isso pode ser alcançado em vários níveis.

Propagação de tradução

Com *Permitir propagação de tradução* habilitada (que é o padrão, consulte *Configuração de componente*), todas as novas traduções são feitas automaticamente em todos os componentes com textos correspondentes. Estas traduções são devidamente creditadas ao usuário que traduz atualmente em todos os componentes.

Nota: A propagação de tradução requer a chave para ser compatível com formatos de tradução monolíngue, por isso tenha isso em mente ao criar chaves de tradução.

Verificação de consistência

A verificação check-inconsistente é acionada sempre que os textos são diferentes. Você pode usar isso para revisar tais diferenças manualmente e escolher a tradução certa.

Tradução automática

A tradução automática com base em diferentes componentes pode ser uma maneira de sincronizar as traduções entre os componentes. Você pode acioná-la manualmente (veja *Tradução automática*) ou fazê-la ser executada automaticamente na atualização do repositório usando uma extensão (veja *Tradução automática*).

2.10 Licenciando traduções

Você pode especificar sob quais traduções de licença são contribuídas. Isso é especialmente importante de se as traduções forem abertas ao público, para estipular para que elas possam ser usadas.

Você deve especificar as informações da licença da *Configuração de componente*. Você deve evitar exigir um contrato de licença de colaborador, embora seja possível.

2.10.1 Informações de licença

Ao especificar informações de licenças (nome da licença e URL), essas informações são mostradas na seção de informações de tradução da respectiva *Configuração de componente*.

Normalmente este é o melhor lugar para publicar informações de licenciamento se nenhum consentimento explícito for necessário. Se o seu projeto ou tradução não for livre, você provavelmente precisa de consentimento prévio.

2.10.2 Acordo de colaborador

Se você especificar um contrato de licença de colaborador, apenas os usuários que concordaram com ele poderão contribuir. Este é um passo claramente visível ao acessar a tradução:

Contribution to this translation requires you to agree with a contributor agreement. [View contributor agreement](#)

Language	Translated	Unfinished	Unfinished words	Unfinished characters	Checks	Suggestions	Comments
Czech 🇨🇪 GPL-3.0	✓						
Hebrew 🇮🇱 GPL-3.0	✓						
Hungarian 🇮🇪 GPL-3.0	81%	4	5	32			
English 🇬🇧 GPL-3.0	✓						

[Start new translation](#)

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

O texto inserido é formatado em parágrafos e links externos podem ser incluídos. A marcação HTML não pode ser usada.

2.10.3 Licenças de usuário

Qualquer usuário pode revisar todas as licenças de tradução de todos os projetos públicos na instância a partir de seu perfil:

Licenses

Please pay attention to the licensing info, as this specifies how translations can be used.

By registering you agree to use your name and e-mail in the commits, and provide your contribution under the license defined by each localization project.

You have agreed to the following as a contributor:

- [WeblateOrg/Language names](#)

Licenses for individual translations

GNU General Public License v3.0 or later [GPL-3.0](#) ⓘ

[WeblateOrg/WeblateOrg](#) [WeblateOrg/Djangojs](#) [WeblateOrg/Django](#) [WeblateOrg/Language names](#)

MIT License [MIT](#) ⓘ

[WeblateOrg/Android](#)

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

2.11 Processo de tradução

2.11.1 Votação de sugestões

Todos podem adicionar sugestões por padrão, para serem aceitas por usuários autenticados. A votação de sugestões pode ser usada para fazer o uso de um texto quando mais de um usuário 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 usuário fazendo a sugestão, se ela for lançada).

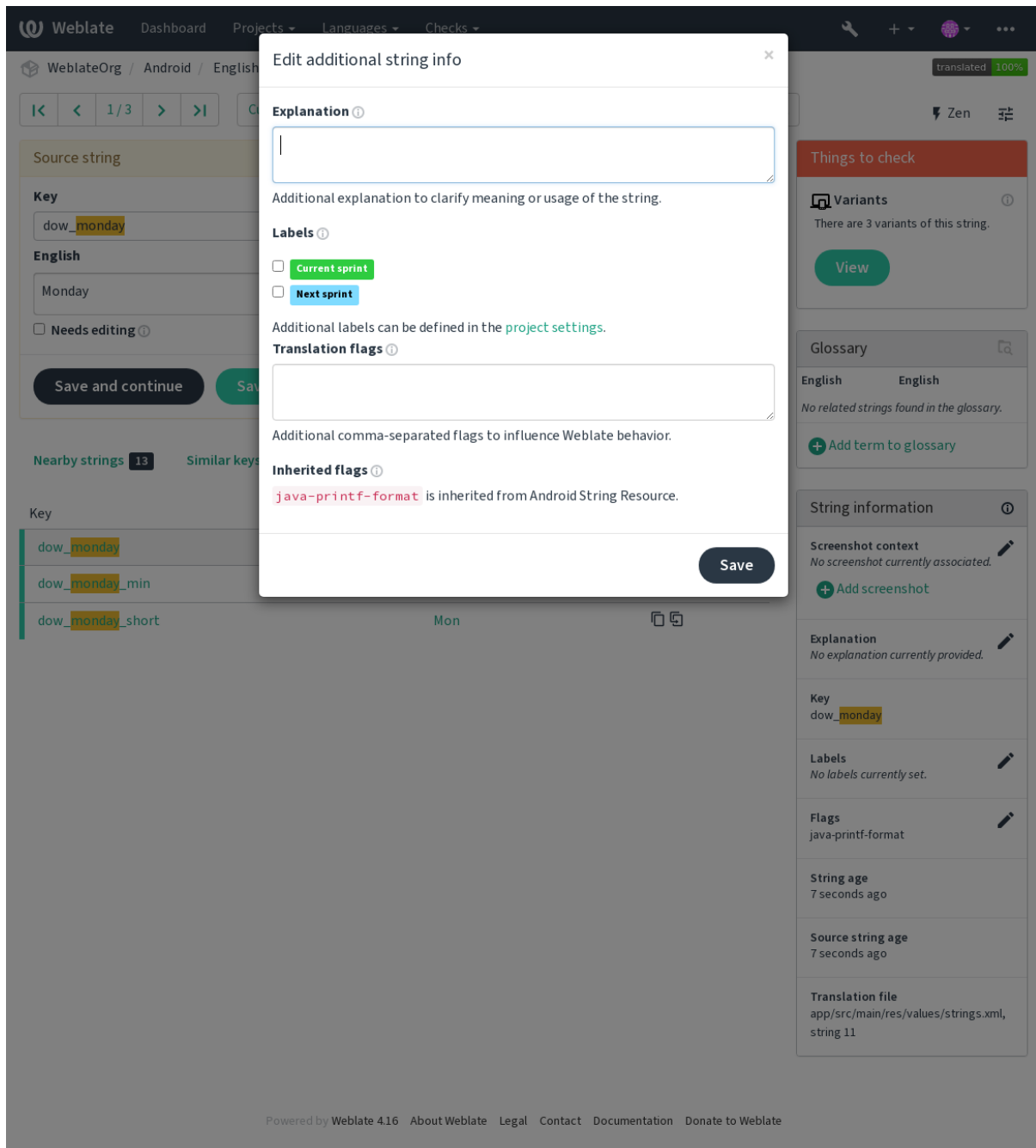
Nota: Uma vez que a aceitação automática está configurada, usuários normais perdem o privilégio de salvar diretamente traduções ou aceitar sugestões. Isso pode ser sobrescrito pela *permissão Editar texto quando as sugestões são forçadas*.

Você pode combiná-las com *controle de acesso* para uma das configurações a seguir:

- Usuários sugerem e votam por sugestões e um grupo limitado controla o que é aceito. - Ativar votação. - Desativar aceitação automática. - Não deixar usuários salvarem traduções.
- Usuários sugerem e votam em sugestões com aceitação automática uma vez que um número definido deles concorda. - Ativar votação. - Definir o número desejado de votos para aceitação automática.
- Voto opcional para sugestões. (Pode ser usado opcionalmente pelos usuários quando não têm certeza sobre uma tradução, marcando múltiplas sugestões.) - Apenas ativar a votação.

2.11.2 Informações adicionais sobre textos 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 arquivos 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 tela* ou *Marcadores*.

Powered by [Webplate 4.16](#) | [About Webplate](#) | [Legal](#) | [Contact](#) | [Documentation](#) | [Donate to Webplate](#)

Priorização dos textos

Novo na versão 2.0.

A prioridade dos textos pode ser alterada 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.

Ver 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 usando marcadores*.

Os marcadores de textos também são herdados dos *Marcadores de tradução* na *Configuração de componente* e marcadores do arquivo de tradução (veja *Formatos de arquivos suportados*).

Ver também:

Verificações de qualidade, *Personalizando comportamento usando 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. Você pode usar Markdown para incluir links e outras marcações.

Contexto visual para textos

Novo na versão 2.9.

Você pode enviar uma captura de tela mostrando 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 tela enviada é mostrada na barra lateral de contexto de tradução:

The screenshot displays the Weblate web interface for managing translations. The top navigation bar includes links for Dashboard, Projects, Languages, and Checks. The main area shows the translation process for a string in Czech. The English source string is "Help text for automatic translation tool". The Czech translation is "Automatický překlad prostřednictvím strojového překladu používá aktivní enginy strojového překladu pro získání nejlepších možných překladů a použije je na tento projekt." The right sidebar contains a glossary with terms like "machine translation" and "project", and a string information panel with details like "String age" and "Source string location".

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Além de *Informações adicionais sobre textos fonte*, as capturas de tela têm uma interface de gerenciamento separada no menu *Ferramentas*. Envie capturas de tela, atribua-as a textos fonte manualmente ou use o reconhecimento ótico de caracteres para fazer isso.

Depois que uma captura de tela é carregada, esta interface lida com o gerenciamento 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.			


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

Image

Currently: [screenshots/screenshot.png](#)

Change:

No file chosen

Upload JPEG or PNG images up to 2000x2000 pixels.

Screenshot details

Created	now
Uploaded by	 testuser
Language	English

Delete screenshot

Deleting screenshot will remove it from all associated source strings.

2.12 Verificações e correções

2.12.1 Correções automáticas personalizadas

Você também pode implementar sua própria correção automática, além das padrão e incluí-las em `AUTOFIX_LIST`.

As correções automáticas são poderosas, mas também podem causar danos; tenha cuidado ao escrever um.

Por exemplo, a correção automática a seguir iria substituir cada ocorrência do texto `foo`, em uma tradução com `bar`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy as _

from weblate.trans.autofixes.base import AutoFix

class ReplaceFooWithBar(AutoFix):
    """Replace foo with bar."""

    name = _("Foobar")

    def fix_single_target(self, target, source, unit):
        if "foo" in target:
            return target.replace("foo", "bar"), True
        return target, False
```

Para instalar verificações personalizadas, forneça um caminho totalmente qualificado para a classe Python em `AUTOFIX_LIST`, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

2.12.2 Personalizando comportamento usando marcadores

Você pode efetuar um ajuste fino do comportamento do Weblate usando marcadores. Isso pode ser feito a nível de texto fonte (veja *Informações adicionais sobre textos fonte*) ou na *Configuração de componente* (*Marcadores de tradução*). Alguns formatos de arquivo também permitem especificar marcadores diretamente no formato (veja *Formatos de arquivos suportados*).

Os marcadores são separados por vírgulas, os parâmetros são separados por caracteres de dois pontos. Você pode usar aspas para incluir espaços em branco ou caracteres especiais no texto. Por exemplo:

```
placeholders:"special:value":"other value", regex:.*
```

Aspas simples e duplas são aceitas, e caracteres especiais são tratados com barra invertida:

```
placeholders:"quoted \"string\"":'single \'quoted\''
```

Aqui está uma lista de marcadores atualmente aceitos:

rst-text

Trata um texto como um documento reStructuredText, afeta *Tradução não alterada*.

dos-eol

Usa marcadores de ponta de linha do DOS em vez dos Unix (`\r\n` em vez de `\n`).

read-only

O texto é somente leitura e não deve ser editado no Weblate, veja *Textos somente leitura*.

priority:N

Prioridade do texto. Os textos de maior prioridade são apresentados primeiro para tradução. A prioridade padrão é 100, a maior prioridade que um texto tem, mais cedo é oferecido para tradução.

max-length:N

Limita o comprimento máximo de um texto a caracteres N, veja *Comprimento máximo da tradução*.

xml-text

Trata o texto como documento XML, afeta *Sintaxe XML* e *Marcação XML*.

font-family:NOME

Define a família de fontes para verificações da renderização, veja *Gerenciando fontes*.

font-weight:PESO

Define o peso da fonte para verificações da renderização, veja *Gerenciando fontes*.

font-size:TAMANHO

Define o tamanho da fonte para verificações da renderização, veja *Gerenciando fontes*.

font-spacing:ESPAÇAMENTO

Define o espaçamento de letras para verificações da renderização, veja *Gerenciando fontes*.

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:...

Textos 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 *Comprimento 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 este texto como uma variante do texto com o fonte correspondente. Veja *variants*.

regex:EXPRESSÃO REGULAR

Expressão regular para corresponder à tradução; veja *Expressão regular*.

forbidden

Indica tradução proibida em um glossário, veja *Traduções proibidas*.

strict-same

Faz com que “Tradução não alterada” evite usar a lista negra de palavras embutidas, veja *Tradução não alterada*.

check-glossary

Habilita a verificação de qualidade *Não segue o glossário*.

angularjs-format

Habilita a verificação de qualidade *Texto de interpolação AngularJS*.

c-format

Habilita a verificação de qualidade *Formato C*.

c-sharp-format

Habilita a verificação de qualidade *Formato C#*.

es-format

Habilita a verificação de qualidade *Literais de modelo de ECMAScript*.

i18next-interpolation

Habilita a verificação de qualidade *Interpolação de i18next*.

icu-message-format

Habilita a verificação de qualidade *MessageFormat do ICU*.

java-printf-format

Habilita a verificação de qualidade *Formato Java*.

java-format

Habilita a verificação de qualidade *MessageFormat do Java*.

javascript-format

Habilita a verificação de qualidade *Formato JavaScript*.

lua-format

Habilita a verificação de qualidade *Formato Lua*.

object-pascal-format

Habilita a verificação de qualidade *Formato de Object Pascal*.

percent-placeholders

Habilita a verificação de qualidade *Espaços reservados de porcentagem*.

perl-format

Habilita a verificação de qualidade *Formato Perl*.

php-format

Habilita a verificação de qualidade *Formato PHP*.

python-brace-format

Habilita a verificação de qualidade *Formato de chaves Python*.

python-format

Habilita a verificação de qualidade *Formato Python*.

qt-format

Habilita a verificação de qualidade *Formato Qt*.

qt-plural-format

Habilita a verificação de qualidade *Formato de plural Qt*.

ruby-format

Habilita a verificação de qualidade *Formato Ruby*.

scheme-format

Habilita a verificação de qualidade *Formato Scheme*.

vue-format

Habilita a verificação de qualidade *Formatação Vue 118n*.

md-text

Trata o texto como um documento Markdown. Habilita as verificações de qualidade afeta *Links Markdown*, *Referências Markdown* e *Sintaxe Markdown*.

case-insensitive

Ajustar o comportamento de verificação para não diferenciar maiúsculas de minúsculas. Atualmente, isso afeta apenas a verificação de qualidade *Espaços reservados*.

safe-html

Habilita a verificação de qualidade *HTML inseguro*.

url

O texto deve consistir apenas em uma URL. Habilita a verificação de qualidade *URL*.

ignore-all-checks

Ignora todas as verificações de qualidade.

ignore-bbcode

Ignora a verificação de qualidade *Marcação de BBCode*.

ignore-duplicate

Ignora a verificação de qualidade *Há palavras duplicadas de forma consecutiva*.

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 *Texto de interpolação AngularJS*.

ignore-c-format

Ignora a verificação de qualidade *Formato C*.

ignore-c-sharp-format

Ignora a verificação de qualidade *Formato C#*.

ignore-es-format

Ignora a verificação de qualidade *Literais de modelo de ECMAScript*.

ignore-i18next-interpolation

Ignora a verificação de qualidade *Interpolação de i18next*.

ignore-icu-message-format

Ignora a verificação de qualidade *MessageFormat do ICU*.

ignore-java-format

Ignora a verificação de qualidade *MessageFormat do Java*.

ignore-java-printf-format

Ignorar a verificação de qualidade *Formato Java*.

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 de Object Pascal*.

ignore-percent-placeholders

Ignora a verificação de qualidade *Espaços reservados de porcentagem*.

ignore-perl-format

Ignora a verificação de qualidade *Formato Perl*.

ignore-php-format

Ignora a verificação de qualidade *Formato PHP*.

ignore-python-brace-format

Ignora a verificação de qualidade *Formato de chaves Python*.

ignore-python-format

Ignora a verificação de qualidade *Formato Python*.

ignore-qt-format

Ignora a verificação de qualidade *Formato Qt*.

ignore-qt-plural-format

Ignora a verificação de qualidade *Formato de plural Qt*.

ignore-ruby-format

Ignora a verificação de qualidade *Formato Ruby*.

ignore-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 usada*.

ignore-md-link

Ignora a verificação de qualidade *Links Markdown*.

ignore-md-reflink

Ignora a verificação de qualidade *Referências Markdown*.

ignore-md-syntax

Ignora a verificação de qualidade *Sintaxe Markdown*.

ignore-max-length

Ignora a verificação de qualidade *Comprimento máximo da tradução*.

ignore-max-size

Ignora a verificação de qualidade *Tamanho máximo da tradução*.

ignore-escaped-newline

Ignora a verificação de qualidade *\n não correspondente*.

ignore-end-colon

Ignora a verificação de qualidade *Caractere de dois pontos não correspondente*.

ignore-end-ellipsis

Ignora a verificação de qualidade *Reticências não correspondentes*.

ignore-end-exclamation

Ignora a verificação de qualidade *Ponto de exclamação não correspondente*.

ignore-end-stop

Ignora a verificação de qualidade *Ponto final não correspondente*.

ignore-end-question

Ignora a verificação de qualidade *Ponto de interrogação não correspondente*.

ignore-end-semicolon

Ignora a verificação de qualidade *Ponto e vírgula não correspondente*.

ignore-newline-count

Ignora a verificação de qualidade *Quebras de linha descasadas*.

ignore-plurals

Ignora a verificação de qualidade *Faltam plurais*.

ignore-placeholders

Ignora a verificação de qualidade *Espaços reservados*.

ignore-punctuation-spacing

Ignora a verificação de qualidade *Espaçamento de pontuação*.

ignore-regex

Ignora a verificação de qualidade *Expressão regular*.

ignore-same-plurals

Ignora a verificação de qualidade *Mesmos plurais*.

ignore-begin-newline

Ignora a verificação de qualidade *Nova linha no início*.

ignore-begin-space

Ignora a verificação de qualidade *Espaços no início*.

ignore-end-newline

Ignora a verificação de qualidade *Linha em branco no final*.

ignore-end-space

Ignora a verificação de qualidade *Espaço no final*.

ignore-same

Ignora a verificação de qualidade *Tradução não alterada*.

ignore-safe-html

Ignora a verificação de qualidade *HTML inseguro*.

ignore-url

Ignora a verificação de qualidade *URL*.

ignore-xml-tags

Ignora a verificação de qualidade *Marcação XML*.

ignore-xml-invalid

Ignora a verificação de qualidade *Sintaxe XML*.

ignore-zero-width-space

Ignora a verificação de qualidade *Espaço com largura zero*.

ignore-ellipsis

Ignora a verificação de qualidade *Reticências*.

ignore-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 com falha*.

ignore-unnamed-format

Ignora a verificação de qualidade *Várias variáveis sem nome*.

ignore-optional-plural

Ignora a verificação de qualidade *Não pluralizado*.

Nota: Geralmente, a regra é chamada de `ignore-*` para qualquer verificação, usando seu identificador, para que você possa usá-la mesmo para suas verificações personalizadas.

Esses marcadores são entendidos tanto nas configurações de *Configuração de componente*, por configurações de textos fonte quanto no próprio arquivo de tradução (por exemplo, no GNU gettext).

2.12.3 Forçando verificações

Novo na versão 3.11.

Você pode configurar uma lista de verificações que não podem ser ignoradas definindo *Verificações forçadas* em *Configuração de componente*. Cada verificação listada não pode ser dispensado na interface do usuário e qualquer texto com falha nesta verificação é marcado como *Precisa de edição* (veja *Estados de tradução*).

Nota: Ativar a imposição de verificação não a ativa automaticamente. A verificação pode ser ativada adicionando o marcador correspondente aos marcadores de texto ou componente.

Ver também:

Informações adicionais sobre textos fonte, Marcadores de tradução

2.12.4 Gerenciando fontes

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 usuário do Weblate.

A verificação *Tamanho máximo da tradução* usada para calcular as dimensões do texto renderizado precisa ser carregada no Weblate e selecionada usando um marcador de tradução (veja *Personalizando comportamento usando marcadores*).

Ferramenta de gerenciamento de fontes do Weblate em *Fonts* sob o menu *Gerenciar* do seu projeto de tradução fornece interface para carregar e gerenciar fontes. As fontes TrueType ou OpenType podem ser carregadas, configurar grupos de fontes e usá-los na verificação.

Os grupos de fontes permitem definir diferentes fontes para diferentes idiomas, o que é normalmente necessário para idiomas não-latinos:

Font group

Name	default-font		
Default font	Source Sans 3 Bold		
Japanese	language override	Droid Sans Fallback Regular	Remove
Korean	language override	Droid Sans Fallback Regular	Remove

Delete

Add language override

Language

Font

Save

Edit font group

Font group name

default-font

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

Default font

Source Sans 3 Bold

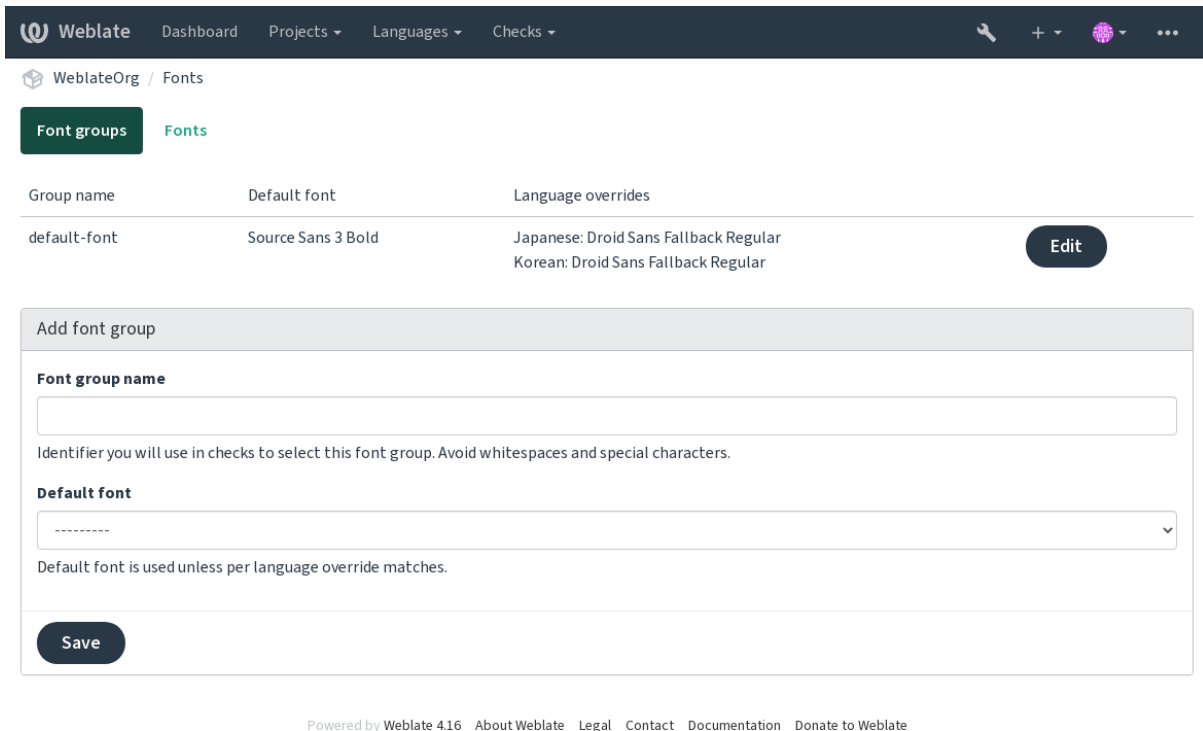
Default font is used unless per language override matches.

Save

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

O grupos de fontes são identificados pelo nome, que não pode conter espaços ou caracteres especiais, de modo que

ele pode ser facilmente utilizado na definição da verificação:



The screenshot shows the Weblate interface for managing fonts. At the top, there's a navigation bar with 'Weblate', 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this, the breadcrumb 'WeblateOrg / Fonts' is visible. The 'Font groups' tab is active, showing a table with one font group: 'default-font' using 'Source Sans 3 Bold' as the default font, with overrides for Japanese and Korean. An 'Edit' button is next to the group. Below the table is a form to 'Add font group' with a text input for the group name, a dropdown for the default font, and a 'Save' button.

Font groups

Group name	Default font	Language overrides
default-font	Source Sans 3 Bold	Japanese: Droid Sans Fallback Regular Korean: Droid Sans Fallback Regular

Edit

Add font group

Font group name

Identifier you will use in checks to select this font group. Avoid whitespaces and special characters.

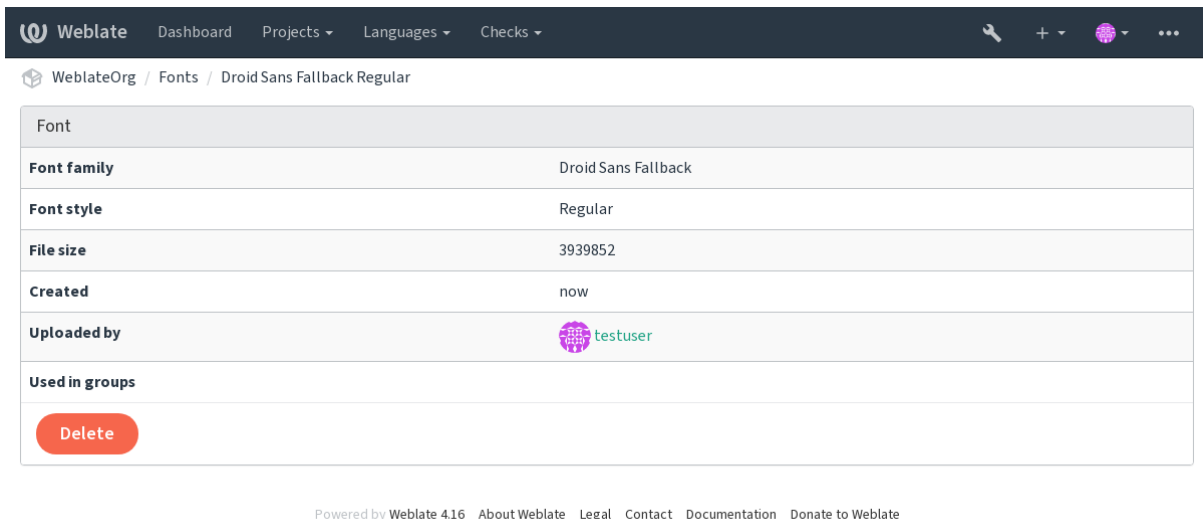
Default font

Default font is used unless per language override matches.

Save

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

A família de fontes e o estilo são automaticamente reconhecidos após carregá-los:



The screenshot shows the details of a font group in Weblate. The breadcrumb is 'WeblateOrg / Fonts / Droid Sans Fallback Regular'. Below this is a table with the following details: Font family (Droid Sans Fallback), Font style (Regular), File size (3939852), Created (now), and Uploaded by (testuser). At the bottom, there's a 'Delete' button.

Font
Font family
Droid Sans Fallback
Font style
Regular
File size
3939852
Created
now
Uploaded by
testuser
Used in groups
Delete

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Você pode ter um número de fontes carregadas para Weblate:

Para usar as fontes para verificar o comprimento do texto, passe-o os marcadores apropriados (veja [Personalizando comportamento usando marcadores](#)). Você provavelmente precisará dos seguintes:

max-size:500

Define o máximo de largura em pixels.

font-family:ubuntu

Define o grupo de fontes para usar especificando seu identificador.

font-size:22

Define o tamanho da fonte em pixels.

2.12.5 Escrevendo as próprias verificações

Uma ampla gama de verificações de qualidade estão embutidas, (veja [Verificações de qualidade](#)), embora eles possam não cobrir tudo o que você deseja verificar. A lista de verificações realizadas pode ser ajustada usando `CHECK_LIST` e você também pode adicionar verificações personalizadas.

1. Crie uma subclasse de `weblate.checks.Check`
2. Defina alguns atributos.
3. Implemente o método `check` (se você quiser lidar com plurais em seu código) ou o método `check_single` (que faz isso por você).

Alguns exemplos:

Para instalar verificações personalizadas, forneça um caminho totalmente qualificado para a classe Python em `CHECK_LIST`, veja [Verificações de qualidade personalizadas, extensões e correções automáticas](#).

Verificando se o texto de tradução não contém “foo”

Esta é uma verificação bastante simples que apenas verifica se a tradução não possui o texto “foo”.

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""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 © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""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
```

(continua na próxima página)


(continuação da página anterior)





```
def check_single(self, source, target, unit):
    """We don't check target strings here."""
    return False
```


2.13 Configurando sugestões automáticas

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:


Dashboard
Projects ▾
Languages ▾
Checks ▾


WeblateOrg / Automatic suggestions

Configured automatic suggestion services ⓘ

There are no services currently installed.

Available automatic suggestion services ⓘ

AWS ⓘ	Install
Amagama ⓘ	Install
Apertium APy ⓘ	Install
Baidu ⓘ	Install
DeepL ⓘ	Install
Glosbe ⓘ	Install
Google Translate ⓘ	Install
Google Translate API v3 ⓘ	Install
IBM ⓘ	Install
LibreTranslate ⓘ	Install
Microsoft Terminology ⓘ	Install
Microsoft Translator ⓘ	Install
ModernMT ⓘ	Install
MyMemory ⓘ	Install
Netease Sight ⓘ	Install
SAP Translation Hub ⓘ	Install
Weblate ⓘ	Install
Weblate Translation Memory ⓘ	Install
Yandex ⓘ	Install
Youdao Zhiyun ⓘ	Install
tmserver ⓘ	Install

Some services will ask for additional configuration during installation.

Nota: Eles estão sujeitos aos seus termos de uso, então certifique-se de ter permissão para usá-los.

Os serviços traduzem do idioma fonte conforme configurado em [Configuração de componente](#), veja [Idioma fonte](#).

Ver também:

[Sugestões automáticas](#)

2.13.1 amaGama

ID do serviço

amagama

Configuração

Este serviço não tem configuração.

Instalação especial do [tmserver](#) executado pelos autores do Virtaal.

Ver também:

[Installing amaGama](#), [Amagama](#), [Memória de tradução do amaGama](#)

2.13.2 Apertium APy

ID do serviço

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 seu próprio servidor Apertium-APy.

Ver também:

[Site do Apertium](#), [Documentação do Apertium APy](#)

2.13.3 AWS

Novo na versão 3.1.

ID do serviço

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 em uma variedade de idiomas compatíveis.

Ver também:

[Documentação do Amazon Translate](#)

2.13.4 Baidu

Novo na versão 3.2.

ID do serviço

baidu

Configuração

key	ID do cliente
secret	Chave secreta

Serviço de tradução de máquina fornecido pelo Baidu.

Este serviço usa uma API e você precisa obter um ID e uma chave de API do Baidu para usá-lo.

Ver também:

[API de tradução do Baidu](#)

2.13.5 DeepL

Novo na versão 2.20.

ID do serviço

deepl

Configuração

url	URL da API
key	Chave da API

DeepL é um serviço pago que fornece boa tradução de máquina para alguns idiomas. Você precisa comprar a assinatura do *DeepL API* ou pode usar o plano legado *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 usuário.

Anteriormente, o Weblate era classificado como uma ferramenta CAT pelo DeepL, por isso deveria usar a API v1, mas agora é entendido que deve usar a API v2. Portanto, seu padrão é v2, e você pode alterar isso para v1 no caso de você ter uma assinatura CAT existente e querer que o Weblate use isso.

A maneira mais fácil de descobrir qual usar é abrir uma URL como a seguinte em 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”, você tem a URL correta; se não, tente os outros três.

O Weblate oferece suporte à formalidade no DeepL, ele escolherá uma correspondência com base no idioma (por exemplo, há `de@formal` e `de@informal`).

Ver também:

[Site do DeepL](#), [Preços do DeepL](#), [Documentação do DeepL](#)

2.13.6 Glosbe

ID do serviço`glosbe`**Configuração***Este serviço não tem configuração.*

Dicionário gratuito e memória de tradução para quase todos os idiomas vivos.

O uso da API é gratuito, mas o uso das traduções está sujeito à licença da fonte de dados usada. Existe um limite de chamadas que podem ser feitas de um IP em um determinado período de tempo, para evitar abusos.

Ver também:

[Site do Glosbe](#)

2.13.7 Google Tradutor

ID do serviço`google-translate`**Configuração**

<code>key</code>	Chave da API
------------------	--------------

Serviço de tradução de máquina fornecido pelo Google.

Este serviço usa a API do Google Tradutor, e você precisa obter uma chave de API e ativar a cobrança no console da API do Google.

Ver também:

[Documentação do Google Tradutor](#)

2.13.8 Google Tradutor API v3

ID do serviço`google-translate-api-v3`**Configuração**

<code>credentials</code>	Informações do serviço de contas do Google Tradutor
<code>project</code>	Projeto do Google Tradutor
<code>location</code>	Regionalização do Google Tradutor

Serviço de tradução de máquina fornecido pelos serviços do Google Cloud.

Ver também:

[Documentação do Google Translate](#), [Autenticar os serviços do Cloud usando bibliotecas de cliente](#), [Criando um projeto Google Translate](#), [Regiões do Google Cloud App Engine](#)

2.13.9 LibreTranslate

Novo na versão 4.7.1.

ID do serviço

libretranslate

Configuração

url	URL da API
key	Chave da 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.

Ver também:

[Website do LibreTranslate](#), [Repositório do LibreTranslate](#), [Espelhos do LibreTranslate](#)

2.13.10 Terminologia Microsoft

Novo na versão 2.19.

ID do serviço

microsoft-terminology

Configuração

Este serviço não tem configuração.

A API do Serviço de Terminologia Microsoft permite que você acesse programaticamente a terminologia, as definições e os textos da interface do usuário (UI) disponíveis no Portal de Idiomas por meio de um serviço da web.

Ver também:

[API do Serviço de Terminologia Microsoft](#)

2.13.11 Microsoft Translator

Novo na versão 2.10.

ID do serviço

microsoft-translator

Configuração

key	Chave da API	
base URL	Escolhas disponíveis:	
base da aplicação	api.cognitive.microsofttranslator.com	– Global (não-regional)
	api-apc.cognitive.microsofttranslator.com	– Ásia Pacífico
	api-eur.cognitive.microsofttranslator.com	– Europa
	api-nam.cognitive.microsofttranslator.com	– América do Norte
	api.translator.azure.cn	– China
	api.cognitive.microsofttranslator.us	– Azure US Government cloud
endpoint	Authentication service URL	Regional or multi-service can be specified using region field below.
		Escolhas disponíveis:
		api.cognitive.microsoft.com – Global
		api.cognitive.azure.cn – China
		api.cognitive.microsoft.us – Azure US Government cloud
region	Authentication service region	

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 você usa com a API V2 do Tradutor pode ser usada com API 3.

API V3 de texto do Tradutor

Você precisa se registrar no portal do Azure e usar a chave obtida lá. Com as novas chaves do Azure, você também precisa definir `region` para a região do seu serviço.

Dica: Para Azure China, use sua extremidade do Portal do Azure.

Ver também:

Serviços Cognitivos - API de Tradução de Texto, Portal do Microsoft Azure <<https://portal.azure.com/>> `_, `URLs base, “Autenticação com um recurso de vários serviços” Seção “Autenticação com um token de acesso”

2.13.12 ModernMT

Novo na versão 4.2.

ID do serviço

modernmt

Configuração

url	URL da API
key	Chave da API

Ver também:

[API do ModernMT](#),

2.13.13 MyMemory

ID do serviço

mymemory

Configuração

email	E-mail de contato
username	Nome de usuário
key	Chave da API

Grande memória de tradução com tradução de máquina.

O uso gratuito e anônimo está atualmente limitado a 100 solicitações/dia ou a 1000 solicitações/dia quando você fornece um endereço de e-mail de contato em ``email``. Você também pode solicitar mais.

Ver também:

[Site do MyMemory](#)

2.13.14 Netease Sight

Novo na versão 3.3.

ID do serviço

netease-sight

Configuração

key	ID do 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.

Ver também:

[Plataforma de Tradução NetEase Sight](#)

2.13.15 SAP Translation Hub

ID do serviço

sap-translation-hub

Configuração

url	URL da API	
key	Chave da API	
user-name	Nome de usuário SAP	
password	Senha SAP	
enable_autom	Ativar tradução automática	
domain	Domínio da tradução	A identificação de um domínio de tradução, por exemplo, BC. Se você não especificar um domínio, o método procura por traduções em todos os domínios disponíveis.

Serviço de tradução de máquina fornecido por SAP.

Você precisa ter uma conta SAP (e o SAP Translation Hub habilitado na SAP Cloud Platform) para usar este serviço.

Você também pode configurar se deseja utilizar serviços de tradução automática, além do base de dados de termos.

Nota: Para acessar a API Sandbox, você precisa definir `url` e `key`.

Para acessar a API produtiva, você precisa definir `url`, `username` e `password`.

Ver também:

[API do SAP Translation Hub](#), [Criar URL de base do SAP Translation Hub](#)

2.13.16 tmserver

ID do serviço

tmserver

Configuração

url	URL da API
-----	------------

Você pode executar seu próprio servidor de memória de tradução usando aquele que vem com o Translate-toolkit e deixar que o Weblate converse com ele. Você também pode usá-lo com um servidor amaGama, que é uma versão aprimorada do tmserver.

1. Primeiro, você vai querer 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 suas solicitações:

```
tmserver -d /var/lib/tm/db
```

- Configure o Weblate para falar com ele, a URL padrão é `http://localhost:8888/tmserver/`.

Ver também:

[tmserver](#) [Installing amaGama](#), [Amagama](#), [Memória de Tradução Amagama](#)

2.13.17 IBM Watson Language Translator

ID do serviço

ibm

Configuração

url	URL da API
key	Chave da API

IBM Watson Language Translator translates text from one language to another. The service offers multiple domain-specific models.

Ver também:

[Watson Language Translator](#), [IBM Cloud API Docs](#)

2.13.18 Weblate

ID do serviço

weblate

Configuração

Este serviço não tem configuração.

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.

2.13.19 Weblate Translation Memory

Novo na versão 2.20.

ID do serviço

weblate-translation-memory

Configuração

Este serviço não tem configuração.

Use [Memória de tradução](#) como um serviço de tradução de máquina. Qualquer texto que tenha sido traduzido no passado (ou carregado na memória de tradução) pode ser traduzido dessa maneira.

2.13.20 Yandex

ID do serviço

yandex

Configuração

key	Chave da 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.

Ver também:

[API do Yandex Tradutor](#), Fornecido por [Yandex.Translate](#)

2.13.21 Youdao Zhiyun

Novo na versão 3.2.

ID do serviço

youdao-zhiyun

Configuração

key	ID do 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.

Ver também:

[Youdao Zhiyun Serviço de Tradução de Linguagem Natural](#)

2.13.22 Tradução da máquina personalizada

Você também pode implementar seus próprios serviços de tradução automática usando algumas linhas de código Python. Este exemplo implementa a tradução automática em uma lista fixa de idiomas usando o módulo Python `dictionary`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""Machine translation example."""

import dictionary

from weblate.machinery.base import MachineTranslation

class SampleTranslation(MachineTranslation):
    """Sample machine translation interface."""

    name = "Sample"
```

(continua na próxima página)

(continuação da página anterior)

```

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}

```


Você pode listar sua própria classe em `WEBLATE_MACHINERY` e o Weblate irá utilizá-la.


2.14 Extensões

Novo na versão 2.19.

Extensões fornecem maneiras para personalizar e automatizar o fluxo de trabalho de tradução. Administradores podem gerenciar extensões a partir do menu *Gerenciar* ↓ *Extensões* dos respectivos componente de tradução.

Dica: Você 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

Ensures a consistent set of languages is used for all components within a project.

project wide


Install


Component discovery

Automatically adds or removes project components based on file changes in the version control system.


repository wide

Install


Bulk edit


Bulkedit flags, labels, or states of strings.

Install


Statistics generator


Generates a file containing detailed info about the translation status.

Install


Prefill translation with source


Fills in translation strings with source string.

Install


Pseudolocale generation


Generates a translation by adding prefix and suffix to source strings automatically.

Install


Contributors in comment


Updates the comment part of the PO file header to include contributor names and years of contributions.

Install


Customize gettext output


Allows customization of gettext output behavior, for example line wrapping.

Install


Generate MO files


Automatically generates a MO file for every changed PO file.

Install


Update PO files to match POT (msgmerge)

Updates all PO files (as configured by "File mask") to match the POT file (as configured by "Template for new translations") using msgmerge.


Install


Squash Git commits

Squash Git commits prior to pushing changes.

repository wide


Install


Stale comment removal

Set a timeframe for removal of comments.

project wide

Install


Stale suggestion removal

Set a timeframe for removal of suggestions.

project wide

Install

Some add-ons will ask for additional configuration during installation.

2.14.1 Extensões embutidas

Tradução automática

Novo na versão 3.9.

ID da extensão

`weblate.autotranslate.autotranslate`

Configuração

<code>mode</code>	Modo de tradução automática	Escolhas disponíveis: <code>suggest</code> – Adiciona como sugestão <code>translate</code> – Adiciona como tradução <code>fuzzy</code> – Adiciona como “Necessita edição”
<code>filter_</code>	Filtro de pesquisa	Observe que a tradução de todos os textos descartará todas as traduções existentes. Escolhas disponíveis: <code>all</code> – Todos os textos <code>nottranslated</code> – Textos não traduzidos <code>todo</code> – Textos não finalizados <code>fuzzy</code> – Textos marcados para edição <code>check: inconsistent</code> – Verificação com falha: Inconsistente
<code>auto_</code>	Fonte das traduções automáticas	Escolhas disponíveis: <code>others</code> – Outros componentes da tradução <code>mt</code> – Tradução de máquina
<code>component</code>	Componentes	Insira o slug de um componente para ser usado como fonte, mantenha em branco para usar todos os componentes no projeto atual.
<code>engine</code>	Mecanismos de tradução de máquina	
<code>threshold</code>	Limite de pontuação	

Gatilhos

atualização de componentes, diário

Traduz automaticamente textos usando tradução de máquina ou outros componentes.

Ela é acionada:

- Quando novos textos aparecem em um componente.
- Uma vez por mês para cada componente, isso pode ser configurado usando `BACKGROUND_TASKS`.

Ver também:

Tradução automática, Mantendo traduções iguais entre componentes

CDN de localização do JavaScript

Novo na versão 4.2.

ID da extensão

`weblate.cdn.cdnjs`

Configuração

<code>th-reshold</code>	Limiar de tradução	Limiar para inclusão de traduções.
<code>css_selector</code>	Seletor CSS	Seletor CSS para detectar elementos localizáveis.
<code>cookie_name</code>	Nome do cookie de idioma	Nome do cookie que armazena preferência de idioma.
<code>files</code>	Extrair textos de arquivos HTML	Lista de nomes de arquivos em URLs atuais ou remotos para analisar para textos traduzíveis.

Gatilhos

diário, pós-commit do repositório, pós-atualização do repositório

Publica traduções para rede de entrega de conteúdo para uso em localização de JavaScript ou de HTML.

Pode ser usado para localizar páginas HTML estáticas ou para carregar a localização no código JavaScript.

Gera um URL exclusivo para seu componente, que pode ser incluído em páginas HTML para localizá-los. Veja `weblate-cdn` para mais detalhes.

Ver também:

`cdn-addon-config`, `weblate-cdn`, `cdn-addon-extract`, `cdn-addon-html`

Remover textos vazios

Novo na versão 4.4.

ID da extensão

`weblate.cleanup.blank`

Configuração

Esta extensão não tem configuração.

Gatilhos

pós-commit do repositório, pós-atualização do repositório

Remove textos sem uma tradução dos arquivos de tradução.

Use isso para não ter textos vazios em arquivos de tradução (por exemplo, se sua biblioteca de localização os exibe como em falta em vez de recorrer ao texto fonte).

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Limpar arquivos de tradução

ID da extensão

`weblate.cleanup.generic`

Configuração

Esta extensão não tem configuração.

Gatilhos

pré-commit do repositório, pós-atualização do repositório

Atualiza todos os arquivos de tradução para corresponder ao arquivo base monolíngue. Para a maioria dos formatos isto significa remover chaves de tradução obsoletas que não estão mais presentes no arquivo base.

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Adição de idiomas em falta

ID da extensão

`weblate.consistency.languages`

Configuração

Esta extensão não tem configuração.

Gatilhos

diário, pós-adição do repositório

Garante que um conjunto consistente de idiomas seja usado para todos os componentes de um projeto.

Os idiomas ausentes são verificados uma vez a cada 24 horas e quando novos idiomas são adicionados no Weblate.

Ao contrário da maioria dos outros, esta extensão afeta todo o projeto.

Dica: Traduza automaticamente os textos recém-adicionados com *Tradução automática*.

Descoberta de componente

ID da extensão

`weblate.discovery.discovery`

Configuração

<code>match</code>	Expressão regular para corresponder aos arquivos de tradução	
<code>file_for</code>	Formato de arquivo	
<code>name_tem</code>	Personalizar o nome do componente	
<code>base_fil</code>	Defina o nome do arquivo base monolíngue	Deixe vazio para arquivos de tradução bilíngues.
<code>new_base</code>	Define o arquivo base para novas traduções	Nome do arquivo usado para criar novas traduções. Para gettext escolha o arquivo .pot.
<code>inter-medi-ate_temp</code>	Arquivo de idioma intermediário	Nome do arquivo de tradução intermediária. Na maioria dos casos, este é um arquivo de tradução fornecido por desenvolvedores e é usado ao criar textos fontes reais.
<code>lan-guage_re-copy_add</code>	Filtro de idioma	Expressão regular para filtrar arquivos de tradução ao examinar por uma máscara de arquivo.
<code>remove</code>	Clone extensões do componente principal para as recém-criadas	
<code>confirm</code>	Remova componentes para arquivos inexistentes	
	Confirmo que a correspondência acima parece correta	

Gatilhos

Pós-atualização do repositório

Adiciona ou remove automaticamente componentes do projeto com base nas alterações de arquivo no sistema de controle de versão.

Acionada toda vez que o VCS é atualizado, de outra forma semelhante ao comando de gerenciamento `import-project`. Desta forma, você pode rastrear vários componentes de tradução dentro de um VCS.

A correspondência é feita usando expressões regulares que permitem uma configuração complexa, mas é necessário algum conhecimento para fazê-la. Alguns exemplos para casos de uso comum podem ser encontrados na seção de ajuda de extensões.

Uma vez que você acertar *Salvar*, uma prévia dos componentes correspondentes será apresentada, de onde você pode verificar se a configuração realmente corresponde às suas necessidades:

WebateOrg / 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 (djangojs)	File mask: weblate/locale/*/LC_MESSAGES/djangojs.po weblate/locale/cs/LC_MESSAGES/djangojs.po (cs) weblate/locale/hu/LC_MESSAGES/djangojs.po (hu) weblate/locale/he/LC_MESSAGES/djangojs.po (he)
Django (django)	File mask: weblate/locale/*/LC_MESSAGES/django.po 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 component
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.
(?P<originalHierarchy>.+)/(?P<component>[^/\.]*)/src/main/resources/ApplicationResource_es.properties P<language>[^/\.]*)\.properties	parent/module1/submodule/src/main/resources/ApplicationResources_fr.properties parent/module1/submodule/src/main/resources/ApplicationResource_es.properties parent/module2/src/main/resources/ApplicationResource_de.properties parent/module2/src/main/resources/ApplicationResource_ro.properties	Multi-module Maven project with Java properties translations

You can use Django template markup in both component name and the monolingual base filename, for example:

{{ component }}

Component filename match

{{ component|title }}

Component filename with upper case first letter

Save

Dica: A extensão de descoberta de componentes usa *URLs internas do Weblate*. É uma maneira conveniente de compartilhar a configuração do VCS entre vários componentes. Os componentes vinculados usam o repositório local do componente principal configurado preenchendo `weblate://project/main-component` no campo *Repositório do código-fonte* (em *Gerenciar* ↓ *Configurações* ↓ *Sistema de controle de versão*) de cada componente. Isso economiza tempo com configuração e recursos do sistema também.

Ver também:

Marcação de modelo

Editor 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	Marcadores de tradução a adicionar	
re-move_flags	Marcadores de tradução a remover	
add_label	Etiquetas a adicionar	
re-move_label	Etiquetas para remover	

Gatilhos

atualização do componente

Edição em massa de marcadores, etiquetas ou estados de cadeias.

Automatize o etiquetamento começando com a consulta de pesquisa `NOT has:label` e adicione etiquetas até que todos os textos tenham todas as etiquetas necessárias. Outras operações automatizadas para metadados Weblate também podem ser feitas.

Exemplos:

Tabela5: Rotula novos textos automaticamente

Consulta de pesquisa	<code>NOT has:label</code>
Etiquetas a adicionar	<code>recente</code>

Tabela6: Marcar como somente leitura todos os textos do changelog de Arquivos de metadados de loja de aplicativos

Consulta de pesquisa	<code>language:en AND key:changelogs/</code>
Marcadores de tradução a adicionar	<code>read-only</code>

Ver também:

Editor em massa, *Personalizando comportamento usando marcadores*, *labels*

Marcar traduções não alteradas como “Necessita edição”

Novo na versão 3.1.

ID da extensão

`weblate.flags.same_edit`

Configuração

Esta extensão não tem configuração.

Gatilhos

pós-criação da unidade

Sempre que um novo texto traduzível é importado do VCS e corresponde a um texto fonte, é marcado como necessitando de edição no Weblate. Especialmente útil para formatos de arquivo que incluem textos fonte para textos não traduzidas.

Dica: Você também pode querer reforçar a verificação *Tradução não alterada* adicionando o marcador `strict-same` aos *Marcadores de tradução*.

Ver também:

Estados de tradução

Marcar novos textos fonte como “Necessita edição”

ID da extensão

`weblate.flags.source_edit`

Configuração

Esta extensão não tem configuração.

Gatilhos

pós-criação da unidade

Sempre que um novo texto fonte for importado do VCS, ele é marcado como necessita edição no Weblate. Desta forma você pode filtrar facilmente e editar os textos fontes escritos pelos desenvolvedores.

Ver também:

Estados de tradução

Marcar novas traduções como “Necessita edição”

ID da extensão

`weblate.flags.target_edit`

Configuração

Esta extensão não tem configuração.

Gatilhos

pós-criação da unidade

Sempre que um novo texto traduzível for importado do VCS, ele é marcado como necessita edição no Weblate. Desta forma você pode filtrar facilmente e editar as traduções criadas pelos desenvolvedores.

Ver também:

Estados de tradução

Gerador de estatísticas

ID da extensão

`weblate.generate.generate`

Configuração

<code>filename</code>	Nome do arquivo gerado
<code>template</code>	Conteúdo do arquivo gerado

Gatilhos

pré-commit do repositório

Gera um arquivo contendo as informações detalhadas sobre o status da tradução.

Você pode usar um modelo do Django, tanto de nome de arquivo e conteúdo, veja markdown para uma descrição detalhada de marcação.

Por exemplo, a geração de um arquivo de resumo para cada tradução:

Nome do arquivo gerado

`locale/{{ language_code }}.json`

Conteúdo

```
{
  "language": "{{ language_code }}",
  "strings": "{{ stats.all }}",
  "translated": "{{ stats.translated }}",
  "last_changed": "{{ stats.last_changed }}",
  "last_author": "{{ stats.last_author }}"
}
```

Ver também:

Marcação de modelo

Preencher tradução com fonte

Novo na versão 4.11.

ID da extensão

`weblate.generate.prefill`

Configuração

Esta extensão não tem configuração.

Gatilhos

atualização de componentes, diário

Preenche textos de tradução com texto fonte.

Todos os textos não traduzidos no componente serão preenchidos com o texto fonte e marcados como necessitando edição. Use isso quando você não puder ter textos vazios nos arquivos de tradução.

Geração de pseudolocalidade

Novo na versão 4.5.

ID da extensão

`weblate.generate.pseudolocale`

Configuração

source	Textos fonte			
target	Tradução alvo	Todos os textos nesta tradução serão substituídos		
prefix	Prefixo de texto fixo			
var_prefix	Prefixo de texto variável			
suffix	Sufixo de texto fixo			
var_suffix	Sufixo de texto variável			
var_multipl	Multiplicador de partes variável	de	Quantas vezes repetir a parte variável dependendo do comprimento do texto fonte.	
include_readonly	Incluir textos somente leitura			

Gatilhos

atualização de componentes, diário

Gera uma tradução adicionando prefixo e sufixo de textos fonte automaticamente.

Pseudolocalidades são úteis para encontrar textos que não estão preparados para localização. Isso é feito alterando todos os textos fonte traduzíveis para facilitar a descoberta de textos inalterados ao executar o aplicativo no idioma da pseudolocalidade.

Também é possível encontrar textos 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 usuário 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.

Os textos serão gerados usando o seguinte padrão:

Prefixo de texto fixo Prefixo de texto variável Texto fonte Sufixo de texto variável Sufixo de texto fixo

Dica: Você pode usar idiomas reais para teste, mas existem pseudolocalidades disponíveis no Weblate - `en_XA` e `ar_XB`.

Dica: Você 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 você 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.

Colaboradores nos comentários

ID da extensão

`weblate.gettext.authors`

Configuração

Esta extensão não tem configuração.

Gatilhos

pré-commit do repositório

Atualiza a parte do comentário do cabeçalho do arquivo PO para incluir nomes de contribuidores e anos de contribuições.

O cabeçalho do arquivo PO terá a seguinte aparência:

```
# Michal Čihař <michal@weblate.org>, 2012, 2018, 2019, 2020.  
# Pavel Borecki <pavel@example.com>, 2018, 2019.  
# Filip Hron <filip@example.com>, 2018, 2019.  
# anonymous <noreply@weblate.org>, 2019.
```

Atualizar variável ALL_LINGUAS no arquivo “configure”

ID da extensão

`weblate.gettext.configure`

Configuração

Esta extensão não tem configuração.

Gatilhos

pós-adicionamento do repositório, diário

Atualiza a variável ALL_LINGUAS em arquivos `configure`, `configure.in` ou `configure.ac`, quando uma nova tradução é adicionada.

Personalizar saída do gettext

ID da extensão

`weblate.gettext.customize`

Configuração

<code>width</code>	Quebra de linhas	Por padrão, o gettext quebra as linhas em 77 caracteres e em novas linhas. Com o parâmetro <code>--no-wrap</code> a quebra é feita apenas em novas linhas.
	Escolhas disponíveis:	
	77	Quebra linhas em 77 caracteres e em novas linhas (padrão do gettext)
	65535	Quebra linhas apenas em novas linhas (como “ <code>gettext --no-wrap</code> ”)
	-1	Sem quebra de linhas

Gatilhos

pós-carregamento de armazenamento

Permite personalizar o comportamento de saída do gettext, por exemplo as quebra de linhas.

Ela oferece as seguintes opções:

- Quebrar linhas em 77 caracteres e em novas linhas
- Quebrar linhas apenas em novas linhas
- Sem quebra de linhas

Nota: Por padrão, o gettext quebra as linhas em 77 caracteres e em novas linhas. Com o parâmetro `--no-wrap`, a quebra só é feita em novas linhas.

Atualizar arquivo LINGUAS

ID da extensão

`weblate.gettext.linguas`

Configuração

Esta extensão não tem configuração.

Gatilhos

pós-adição do repositório, diário

Atualiza o arquivo LINGUAS quando uma nova tradução for adicionada.

Gerar arquivos MO

ID da extensão

`weblate.gettext.mo`

Configuração

<code>path</code>	Caminho do arquivo MO gerado	Se não for especificado, o local do arquivo PO será usado.
-------------------	------------------------------	--

Gatilhos

pré-commit do repositório

Gera automaticamente um arquivo MO para cada arquivo PO alterado.

O local do arquivo MO gerado pode ser personalizada e o campo para ele usa *Marcação de modelo*.

Atualizar arquivos PO para corresponder ao POT (msgmerge)

ID da extensão

`weblate.gettext.msgmerge`

Configuração

<code>previous</code>	Manter msgids anteriores de textos traduzidos
<code>no_location</code>	Remove locais de textos traduzidos
<code>fuzzy</code>	Usar correspondência aproximada

Gatilhos

Pós-atualização do repositório

Atualiza todos os arquivos PO (configurados por *Máscara de arquivo*) para corresponder ao arquivo POT (configurado por *Modelo para novas traduções*) usando `msgmerge`.

Acionada sempre que novas alterações forem trazidas do repositório upstream. A maioria das opções de linha de comando do msgmerge podem ser definidas por meio da configuração da extensão.

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Squash de commits git

ID da extensão

`weblate.git.squash`

Configuração

<code>squash</code>	Squash de commits	Escolhas disponíveis: all – Todos os commits em um só language – Por idioma file – Por arquivo author – Por autor
<code>ap- pend_t</code>	Anexar linhas finaliza- doras à mensagem do commit resultado de squash	As linhas finalizadoras são linhas semelhantes aos cabe- çalhos de e-mail RFC 822, no final da parte livre de uma mensagem de commit, como “Co-authored-by: ...”.
<code>com- mit_me</code>	Mensagem de commit	Esta mensagem de commit será usada em vez das men- sagens de commit combinadas a partir de commits após squash.

Gatilhos

pós-commit do repositório

Fazer squash de commits Git antes de fazer push das alterações.

Commits Git podem ter squash feito antes de fazer o push das alterações em um dos seguintes modos:

- Todos os commits em um só
- Por idioma
- Por arquivo
- Por autor

As mensagens de compromisso originais são mantidas, mas a autoria é perdida a menos que *Por autor* seja selecionada ou a mensagem de compromisso seja personalizada para incluí-la.

As mensagens de commit originais podem opcionalmente ser substituídas por uma mensagem de commit personali-
zada.

Linhas finalizadoras (linhas de commits como `Co-authored-by: ...`) podem opcionalmente ser removidas das mensagens de commit originais e anexadas ao final da mensagem de compromisso após um squash. Isso também gera crédito próprio `Co-authored-by:` para cada tradutor.

Personalizar saída JSON

ID da extensão

`weblate.json.customize`

Configuração

<code>sort_keys</code>	Ordenar JSON	chaves
<code>indent</code>	Recuo do JSON	
<code>style</code>	Estilo de recuo do JSON	Escolhas disponíveis: spaces – Espaços tabs – Tabulações

Gatilhos

pós-carregamento de armazenamento

Permite ajustar o comportamento de saída do JSON, por exemplo recuo ou classificação.

Formatar o arquivo de propriedades Java

ID da extensão

`weblate.properties.sort`

Configuração

Esta extensão não tem configuração.

Gatilhos

pré-commit do repositório

Formata e ordena o arquivo de propriedades Java.

- Consolida quebras de linhas para Unix.
- Formatação em maiúsculas de sequências de escape Unicode (caso estejam presentes).
- Tira linhas em branco e comentários.
- Ordena os textos pelas chaves.
- Descarta textos duplicados.

Remoção de comentário obsoleto

Novo na versão 3.7.

ID da extensão

`weblate.removal.comments`

Configuração

age	Dias para manter
-----	------------------

Gatilhos

diário

Define um período de tempo até a remoção de comentários.

Isso pode ser útil para remover comentários antigos que podem ter ficado desatualizados. Use com cuidado, pois comentários ficarem velhos não significam que eles perderam sua importância.

Remoção de sugestão obsoleta

Novo na versão 3.7.

ID da extensão

`weblate.removal.suggestions`

Configuração

age	Dias para manter	
votes	Limite de votação	Limite para remoção. Este campo não tem efeito com a votação desativada.

Gatilhos

diário

Define um período de tempo até 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 em um determinado período de tempo.

Atualizar arquivos RESX

Novo na versão 3.9.

ID da extensão

`weblate.resx.update`

Configuração

Esta extensão não tem configuração.

Gatilhos

Pós-atualização do repositório

Atualiza todos os arquivos de tradução para corresponder ao arquivo base monolíngue do upstream. Textos não usados são removidos e novos são adicionados como cópias do texto fonte.

Dica: Use [Limpar arquivos de tradução](#) se você só quiser remover chaves de tradução obsoletas.

Ver também:

O Weblate atualiza arquivos de tradução além de traduções?

Personalizar saída YAML

Novo na versão 4.15.

ID da extensão

`weblate.yaml.customize`

Configuração

<code>closing_tags</code>	Incluir tag de fechamento para tags XML em branco
---------------------------	---

Gatilhos

pós-carregamento de armazenamento

Permite ajustar o comportamento da saída XML, por exemplo, fechar tags em vez de tags de fechamento automático para tags vazias.

Personalizar saída YAML

Novo na versão 3.10.2.

ID da extensão

`weblate.yaml.customize`

Configuração

indent	Recuo do YAML	
width	Quebra 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_	Quebra de linhas	Escolhas disponíveis: dos – DOS (\r\n) unix – UNIX (\n) mac – MAC (\r)

Gatilhos

pós-carregamento de armazenamento

Permite ajustar o comportamento de saída do YAML, por exemplo comprimento de linha e novas linhas.

2.14.2 Personalizando a lista de extensões

A lista de extensões é configurada por `WEBLATE_ADDONS`. Para adicionar outra extensão, basta incluir o nome absoluto da classe nesta configuração.

2.14.3 Escrevendo extensões

Você pode escrever suas próprias extensões também. Crie uma subclasse de `weblate.addons.base.BaseAddon` para definir os metadados da extensão e, então, implemente uma função de retorno que vai fazer o processamento.

Ver também:*Desenvolvendo extensões*

2.14.4 Escrevendo scripts para extensões

Extensões também podem ser usadas para executar scripts externos. Isso costumava estar integrado no Weblate, mas agora você tem que escrever algum código para embrulhar seu script com uma extensão.

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

"""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
```

(continua na próxima página)

(continuação da página anterior)

```
name = "weblate.example.pre"
# Verbose name and long description
verbose = _("Execute script before commit")
description = _("This add-on executes a script.")

# Script to execute
script = "/bin/true"
# File to add in commit (for pre commit event)
# does not have to be set
add_file = "po/{{ language_code }}.po"
```

Para instruções de instalação, veja *Verificações de qualidade personalizadas, extensões e correções automáticas*.

O script é executado com o diretório atual definido para a raiz do repositório VCS para qualquer componente.

Além disso, estão disponíveis as seguintes variáveis de ambiente:

WL_VCS

Sistema de controle de versão usado.

WL_REPO

URL do repositório upstream.

WL_PATH

Caminho absoluto para o repositório VCS.

WL_BRANCH

Novo na versão 2.11.

Ramo do repositório configurado no componente atual.

WL_FILEMASK

Máscara de arquivo para componente atual.

WL_TEMPLATE

Nome de arquivo de modelo para traduções monolíngues (pode estar vazio).

WL_NEW_BASE

Novo na versão 2.14.

Nome de arquivo do arquivo usado para criar novas traduções (pode estar vazio).

WL_FILE_FORMAT

Formato de arquivo usado no componente atual.

WL_LANGUAGE

Idioma da tradução processada atualmente (não disponível para ganchos de nível de componente).

WL_PREVIOUS_HEAD

HEAD anterior após atualização (disponível apenas ao executar o gancho de pós-atualização).

WL_COMPONENT_SLUG

Novo na versão 3.9.

Slug do componente usado para construir a URL.

WL_PROJECT_SLUG

Novo na versão 3.9.

Slug de projeto usado para construir a URL.

WL_COMPONENT_NAME

Novo na versão 3.9.

Nome de componente.

WL_PROJECT_NAME

Novo na versão 3.9.

Nome do projeto.

WL_COMPONENT_URL

Novo na versão 3.9.

URL do componente.

WL_ENGAGE_URL

Novo na versão 3.9.

URL de engajamento do projeto.

Ver também:

Configuração de componente

Processamento de repositório pós-atualização

Pode ser usado para atualizar arquivos de tradução quando a fonte VCS do upstream alterar. Para conseguir isso, lembre-se que o Weblate só vê arquivos dos quais se fez commit com o VCS, então você precisa fazer commit das alterações como parte do script.

Por exemplo, com Gulp, você pode fazê-lo usando o seguinte código:

```
#!/bin/sh
gulp --gulpfile gulp-i18n-extract.js
git commit -m 'Update source strings' src/languages/en.lang.json
```

Processamento pré-commit de traduções

Use o script de commit para alterar automaticamente uma tradução antes de fazer commit dela para o repositório.

Ele é passado como um parâmetro único que consiste o nome de uma tradução atual.

2.15 Memória de tradução

Novo na versão 2.20.

O Weblate vem com uma memória de tradução integrada que consiste no seguinte:

- Memória de tradução importada manualmente (veja *Interface de usuário*).
- 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*.

Para dicas de instalação, veja *Weblate Translation Memory*, que é ativado por padrão.

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 arquivo 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 usuários e projetos.

Memória de tradução por usuário

Armazena todas as traduções do usuário automaticamente na memória de tradução pessoal de cada respectivo usuário.

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 em uma 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 Gerenciando memória de tradução

Interface de usuário

Novo na versão 3.2.

Na interface de usuário básica, você pode gerenciar memórias de tradução por usuário e por projeto. Ela pode ser usada para baixar, limpar ou importar memória de tradução.

Dica: A memória de tradução em JSON pode ser importada para o Weblate, o TMX é fornecido para interoperabilidade com outras ferramentas.

Ver também:

Esquema de memória de tradução do Weblate

Interface de gerenciamento

Existem vários comandos de gerenciamento 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 arquivos TMX ou JSON para a memória de tradução

2.16 Configuração

Todas as configurações estão armazenadas em `settings.py` (como é habitual no Django).

Nota: Após alterar qualquer uma dessas configurações, você precisa reiniciar o Weblate - tanto os processos WSGI quanto os Celery.

Caso seja executado como `mod_wsgi`, você precisa reiniciar o Apache para recarregar a configuração.

Ver também:

Verifique também a [documentação do Django](#) para parâmetros de configuração do próprio Django.

2.16.1 AKISMET_API_KEY

O Weblate pode usar o Akismet para verificar sugestões recebidas anonimamente para spam. Visite akismet.com para comprar uma chave API e associá-la a um site.

2.16.2 ANONYMOUS_USER_NAME

O nome de usuário dos usuários não autenticados.

Ver também:

Controle de acesso

2.16.3 AUDITLOG_EXPIRY

Novo na versão 3.6.

Quantos dias o Weblate deve manter registros de auditoria, que contêm informações sobre a atividade da conta.

O padrão é 180 dias.

2.16.4 AUTH_LOCK_ATTEMPTS

Novo na versão 2.14.

Número máximo de tentativas de autenticação com falha antes da aplicação da limitação de taxa.

Atualmente, isso é aplicado nos seguintes locais:

- Login. Exclui a senha da conta, impedindo que o usuário entre sem solicitar uma nova senha.
- Redefinição de senha. Impede que novos e-mails sejam enviados, evitando o envio de spam aos usuários com muitas tentativas de redefinição de senha.

O padrão é 10.

Ver também:

Limitação de taxa

2.16.5 AUTO_UPDATE

Novo na versão 3.2.

Alterado na versão 3.11: A opção original de ligar/desligar foi alterada para diferenciar quais textos são aceitos.

Atualiza todos repositórios diariamente.

Dica: Útil se você não estiver usando *Ganchos de notificação* para atualizar automaticamente os repositórios do Weblate.

Nota: Existem opções de ligar/desligar, além da seleção de textos para compatibilidade com versões anteriores.

As opções são:

"none"

Sem atualizações diárias.

"remote" e também False

Atualiza apenas os repositórios remotos.

"full" e também True

Atualiza os repositórios remotos e mescla os cópia de trabalho.

Nota: Isso requer que *Tarefas de fundo usando Celery* esteja funcionando e entrará em vigor após ser reiniciado.

2.16.6 AVATAR_URL_PREFIX

Prefixo para construção de URLs de avatars: `${AVATAR_URL_PREFIX}/avatar/${MAIL_HASH}?${PARAMS}`. Os seguintes serviços funcionam:

Gravatar (padrão), conforme <https://gravatar.com/>

```
AVATAR_URL_PREFIX = 'https://www.gravatar.com/'
```

Libravatar, conforme <https://www.libravatar.org/>

```
AVATAR_URL_PREFIX = 'https://www.libravatar.org/'
```

Ver também:

Cache de avatares, *ENABLE_AVATARS*, *Avatares*

2.16.7 AUTH_TOKEN_VALID

Novo na versão 2.14.

Por quanto tempo o token de autenticação e a senha temporária dos e-mails de redefinição de senha são válidos. Definido em número de segundos, usando como padrão 172800 (2 dias).

2.16.8 AUTH_PASSWORD_DAYS

Novo na versão 2.15.

How many days will Weblate reject reusing previously used password for an user.

The checking is based on the audit log, *AUDITLOG_EXPIRY* needs to be at least same as this.

Nota: Mudanças de senha feitas anteriormente ao Weblate 2.15 não serão consideradas para essa política.

O padrão é 180 dias.

2.16.9 AUTOFIX_LIST

Lista de correções automáticas para aplicar ao salvar um texto.

Nota: Forneça um caminho totalmente qualificado para a classe Python que implementa a interface de correção automática.

Correções disponíveis:

`weblate.trans.autofixes.whitespace.SameBookendingWhitespace`

corresponde o espaço em branco no início e no fim do texto com a fonte.

`weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis`

Substitui pontos ao final (...) se o texto 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 dos textos marcados como `safe-html` (veja *HTML inseguro*).

Você pode selecionar quais usar:

```
AUTOFIX_LIST = (  
    "weblate.trans.autofixes.whitespace.SameBookendingWhitespace",  
    "weblate.trans.autofixes.chars.ReplaceTrailingDotsWithEllipsis",  
)
```

Ver também:

Correções automáticas, Correções automáticas personalizadas

2.16.10 BACKGROUND_TASKS

Novo na versão 4.5.2.

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 usuários 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 usuários não privilegiados a adicionar idiomas indesejados. Os administradores do projeto ainda são apresentados com seleção completa de idiomas definidos no Weblate.

Nota: Isso não define novos idiomas para Weblate, ele filtra apenas os existentes no banco de dados.

Exemplo:

```
BASIC_LANGUAGES = {"cs", "it", "ja", "en"}
```

Ver também:

Definições de idioma

2.16.12 BORG_EXTRA_ARGS

Novo na versão 4.9.

Você pode passar argumentos adicionais para **borg create** quando backups embutidos forem acionados.

Exemplo:

```
BORG_EXTRA_ARGS = ["--exclude", "vcs/"]
```

Ver também:

Fazendo backup e movendo o Weblate, `borg create`

2.16.13 CACHE_DIR

Novo na versão 4.16.

Directory where Weblate stores cache files. Defaults to `cache` subfolder in `DATA_DIR`.

Change this to local or temporary filesystem if `DATA_DIR` is on a network filesystem.

O contêiner do Docker usa um volume separado para isso, veja *Volumes de contêiner Docker*.

2.16.14 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 habilitadas com serviços de terceiros (Matomo, Google Analytics, Sentry, ...).

Todos esses tem como padrão uma lista vazia.

Exemplo:

```
# Enable Cloudflare Javascript optimizations
CSP_SCRIPT_SRC = ["ajax.cloudflare.com"]
```

Ver também:

Política de segurança de conteúdo, Políticas de Segurança de Conteúdo (CSP)

2.16.15 CHECK_LIST

Lista de verificações de qualidade para realizar em uma tradução.

Nota: Forneça um caminho totalmente qualificado para a classe Python implementando a interface de verificação.

Ajuste a lista de verificações para incluir as relevantes para você.

Todas as *Verificações de qualidade* embutidas estão ativadas por padrão, de onde você pode alterar essas configurações. Por padrão, eles são comentados em *Amostra de configuração* para que os valores padrão sejam usados. Novas verificações então realizadas para cada nova versão do Weblate.

Você pode desativar todas as verificações:

```
CHECK_LIST = ()
```

Você pode ativar apenas algumas:

```
CHECK_LIST = (  
    "weblate.checks.chars.BeginNewlineCheck",  
    "weblate.checks.chars.EndNewlineCheck",  
    "weblate.checks.chars.MaxLengthCheck",  
)
```

Nota: Alterar esta configuração afeta apenas as traduções recém-alteradas, as verificações existentes ainda serão armazenadas no banco de dados. Para também aplicar alterações nas traduções armazenadas, execute `update-checks`.

Ver também:

Verificações de qualidade, Personalizando comportamento usando marcadores

2.16.16 COMMENT_CLEANUP_DAYS

Novo na versão 3.6.

Exclui comentários após um determinado número de dias. O padrão é `None`, ou seja, nenhuma exclusão.

2.16.17 COMMIT_PENDING_HOURS

Novo na versão 2.10.

Número de horas entre fazer o commit de alterações pendentes por meio da tarefa de segundo plano.

Ver também:

Configuração de componente, Idade das alterações para fazer commit, Executando tarefas de manutenção, commit_pending

2.16.18 CONTACT_FORM

Novo na versão 4.6.

Configura como o e-mail do formulário de contato está sendo 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*. Seu servidor de e-mail precisa permitir o envio de tais e-mails.

2.16.19 DATA_DIR

A pasta na qual Weblate armazena todos os dados. Ela contém links para repositórios VCS, um índice de texto completo e vários arquivos de configuração para ferramentas externas.

Os seguintes subdiretórios geralmente existem:

home

Diretório pessoal usado para invocar scripts.

ssh

Chaves e configuração de SSH.

static

Local padrão para arquivos Django estáticos, especificados por `STATIC_ROOT`. Veja *Servindo arquivos estáticos*.

O contêiner do Docker usa um volume separado para isso, veja *Volumes de contêiner Docker*.

media

Local padrão para arquivos de mídia Django, especificado por `MEDIA_ROOT`. Contém capturas de tela enviadas, veja *Contexto visual para textos*.

vcs

Repositórios de controle de versão para traduções.

backups

Dados de backup diário. Confira *Dados despejados para os backups* para detalhes.

fonts:

Fontes enviadas pelo usuário, veja *Gerenciando fontes*.

cache

Various caches, can be placed elsewhere using `CACHE_DIR`.

O contêiner do Docker usa um volume separado para isso, veja *Volumes de contêiner Docker*.

Nota: Este diretório tem que ser escrito pelo Weblate. Executá-lo como uWSGI significa que o usuário `www-data` deve ter acesso de escrita.

A maneira mais fácil de conseguir isso é fazer do usuário o proprietário do diretório:

```
sudo chown www-data:www-data -R $DATA_DIR
```

O padrão é `/home/weblate/data`, mas espera-se que seja configurado.

Ver também:

Permissões do sistema de arquivos, Fazendo backup e movendo o Weblate, `CACHE_DIR`

2.16.20 DATABASE_BACKUP

Novo na versão 3.1.

Se os backups de banco de dados devem ser armazenados como texto simples, compactado ou ignorado. Os valores autorizados são:

- "plain"
- "compressed"
- "none"

Ver também:

Fazendo backup e movendo o Weblate

2.16.21 DEFAULT_ACCESS_CONTROL

Novo na versão 3.3.

A configuração padrão de controle de acesso para novos projetos:

0

Público

1

Protegido

100

Privado

200

Personalizado

Use *Personalizado* se você está gerenciando a ACL manualmente, o que significa não confiar no gerenciamento interno do Weblate.

Ver também:

Controle de acesso do projeto, Controle de acesso

2.16.22 DEFAULT_AUTO_WATCH

Novo na versão 4.5.

Configura se *Observar automaticamente projetos em contribuição* deve ser ativado para novos usuários. O padrão é `True`.

Ver também:

Notificações

2.16.23 DEFAULT_RESTRICTED_COMPONENT

Novo na versão 4.1.

O valor padrão para restrição de componentes.

Ver também:

Acesso restrito, Escopo de equipes

2.16.24 DEFAULT_ADD_MESSAGE, DEFAULT_ADDON_MESSAGE, DE- FAULT_COMMIT_MESSAGE, DEFAULT_DELETE_MESSAGE, DE- FAULT_MERGE_MESSAGE

Enviar mensagens padrão para diferentes operações, consulte *Configuração de componente* para detalhes.

Ver também:

Marcação de modelo, Configuração de componente, Mensagens de commit, add, merge, extensão e merge request

2.16.25 DEFAULT_ADDONS

Extensões padrão 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",
    },
}
```

Ver também:

install_addon, *Extensões*, *WEBLATE_ADDONS*

2.16.26 DEFAULT_COMMITTER_EMAIL

Novo na versão 2.4.

Endereço de e-mail do committer com o padrão `noreply@weblate.org`.

Ver também:

DEFAULT_COMMITTER_NAME

2.16.27 DEFAULT_COMMITTER_NAME

Novo na versão 2.4.

Nome do committer com o padrão `Weblate`.

Ver também:

DEFAULT_COMMITTER_EMAIL

2.16.28 DEFAULT_LANGUAGE

Novo na versão 4.3.2.

Idioma fonte padrão para usar, por exemplo, em *Idioma fonte*.

O padrão é *en*. O objeto de idioma correspondente precisa existir no banco de dados.

Ver também:

Definições de idioma, *Idioma fonte*

2.16.29 DEFAULT_MERGE_STYLE

Novo na versão 3.4.

Mescla o estilo para quaisquer novos componentes.

- *rebase* - padrão
- *merge*

Ver também:

Configuração de componente, Estilo de mesclagem

2.16.30 DEFAULT_SHARED_TM

Novo na versão 3.2.

Configura o valor padrão de *Usar memória de tradução compartilhada* e `project-contrib_shared_tm`.

2.16.31 DEFAULT_TRANSLATION_PROPAGATION

Novo na versão 2.5.

Configuração padrão para propagação de tradução, sendo o padrão `True`.

Ver também:

Configuração de componente, Permitir propagação de tradução

2.16.32 DEFAULT_PULL_MESSAGE

Configura o título e a mensagem padrão para pull requests.

2.16.33 ENABLE_AVATARS

Se deve-se ativar avatares baseados em Gravatar para os usuários. Por padrão, isto está ativado.

Avatares são buscados e armazenados em cache no servidor, diminuindo o risco de vazamento de informações privadas, acelerando a experiência do usuário.

Ver também:

Cache de avatares, AVATAR_URL_PREFIX, Avatares

2.16.34 ENABLE_HOOKS

Se deve-se habilitar ganchos remotos anônimos.

Ver também:

Ganchos de notificação

2.16.35 ENABLE_HTTPS

Se deve-se enviar links para Weblate como HTTPS ou HTTP. Esta configuração afeta os e-mails enviados e as URLs absolutas geradas.

Na configuração padrão, este também é usado para várias configurações de Django relacionadas ao HTTPS - ele habilita cookies seguros, ativa/desativa HSTS ou habilita redirecionamento para URL HTTPS.

O redirecionamento HTTPS pode ser problemático em alguns casos e você pode ter problemas com o redirecionamento infinito no caso de estar usando um proxy reverso fazendo o encerramento SSL que não passa corretamente os cabeçalhos de protocolo para o Django. Ajuste 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 status SSL.

Ver também:

`SESSION_COOKIE_SECURE,` `CSRF_COOKIE_SECURE,` `SECURE_SSL_REDIRECT,` `SECURE_PROXY_SSL_HEADER` *Definir domínio correto do site*

2.16.36 ENABLE_SHARING

Ativa/desativa o menu *Compartilhar* para que os usuários possam compartilhar o progresso da tradução nas redes sociais.

2.16.37 EXTRA_HTML_HEAD

Novo na versão 4.15.

Insira marcações adicionais no cabeçalho HTML. Pode ser usado para verificação da propriedade do site, por exemplo:

```
EXTRA_HTML_HEAD = '<link href="https://fosstodon.org/@weblate" rel="me">'
```

Aviso: Nenhuma sanitização é executada no texto, ele é inserido como está no cabeçalho HTML.

2.16.38 GET_HELP_URL

Novo na versão 4.5.2.

URL onde o suporte para sua instância do Weblate pode ser encontrado.

2.16.39 GITEA_CREDENTIALS

Novo na versão 4.12.

Lista para credenciais para servidores Gitea.

```
GITEA_CREDENTIALS = {
    "try.gitea.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "gitea.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

Ver também:

Pull requests do Gitea, *Creating a Gitea personal access token*

2.16.40 GITLAB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores GitLab.

```
GITLAB_CREDENTIALS = {
  "gitlab.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "gitlab.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Ver também:

Merge requests do GitLab, *GitLab: Personal access token*

2.16.41 GITHUB_CREDENTIALS

Novo na versão 4.3.

Lista para credenciais para servidores GitHub.

```
GITHUB_CREDENTIALS = {
  "api.github.com": {
    "username": "weblate",
    "token": "your-api-token",
  },
  "github.example.com": {
    "username": "weblate",
    "token": "another-api-token",
  },
}
```

Ver também:

Pull requests do GitHub, *Creating a GitHub personal access token*

2.16.42 BITBUCKETSERVER_CREDENTIALS

Novo na versão 4.16.

List for credentials for Bitbucket servers.

```
BITBUCKETSERVER_CREDENTIALS = {
  "git.self-hosted.com": {
    "username": "weblate",
    "token": "http-access-token",
  },
}
```

Ver também:

Bitbucket Server pull requests, *Bitbucket: HTTP access token*

2.16.43 GOOGLE_ANALYTICS_ID

Google Analytics ID para ativar o monitoramento do Weblate usando o Google Analytics.

2.16.44 HIDE_REPO_CREDENTIALS

Ocultar credenciais de repositório da interface web. No caso de você ter URL do repositório com usuário e senha, o Weblate irá ocultá-la quando as informações relacionadas são mostradas aos usuários.

Por exemplo, em vez de `https://usuário:senha@git.example.com/repo.git`, ela vai mostrar apenas `"https://git.example.com/repo.git"`. Ela tenta limpar mensagens de erro VCS também de forma semelhante.

Nota: Isso está ativado por padrão.

2.16.45 HIDE_VERSION

Novo na versão 4.3.1.

Esconde informação de versão de usuários não autenticados. Isso também faz todos os links de documentação apontar para a última versão, ao invés da versão de combinação da documentação instalada atualmente.

Ocultar versão é uma prática de segurança recomendada em algumas empresas, mas não prevê invasores de descobri a versão sondando o comportamento.

Nota: Isso está desligado por padrão.

2.16.46 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: O valor padrão permite que os usuários financiem o próprio Weblate.

2.16.47 IP_BEHIND_REVERSE_PROXY

Novo na versão 2.14.

Indica se o Weblate está sendo usado atrás de um proxy reverso.

Se definido como `True`, o Weblate obtém endereço IP de um cabeçalho definido por `IP_PROXY_HEADER`.

Aviso: Certifique-se de que você está realmente usando um proxy reverso e que ele define este cabeçalho, caso contrário, os usuários poderão falsificar o endereço IP.

Nota: Isso não está ativado por padrão.

Ver também:

Executando por trás de um proxy reverso, Limitação de taxa, [IP_PROXY_HEADER](#), [IP_PROXY_OFFSET](#)

2.16.48 IP_PROXY_HEADER

Novo na versão 2.14.

Indica de qual cabeçalho o Weblate deve obter o endereço IP quando [IP_BEHIND_REVERSE_PROXY](#) está ativado.

Padrão é `HTTP_X_FORWARDED_FOR`.

Ver também:

Executando por trás de um proxy reverso, Limitação de taxa, [SECURE_PROXY_SSL_HEADER](#), [IP_BEHIND_REVERSE_PROXY](#), [IP_PROXY_OFFSET](#)

2.16.49 IP_PROXY_OFFSET

Novo na versão 2.14.

Indica qual parte de [IP_PROXY_HEADER](#) é usada como endereço IP do cliente.

Dependendo da configuração, este cabeçalho pode consistir em vários endereços IP (por exemplo, `X-Forwarded-For: a, b, client-ip`) e você pode configurar qual endereço do cabeçalho é usado como endereço IP do cliente aqui.

Aviso: Configurar isso afeta a segurança da sua instalação, então você só deve configurá-la para usar proxies confiáveis para determinar o endereço IP.

O padrão é 0.

Ver também:

Executando por trás de um proxy reverso, Limitação de taxa, [SECURE_PROXY_SSL_HEADER](#), [IP_BEHIND_REVERSE_PROXY](#), [IP_PROXY_HEADER](#)

2.16.50 LEGAL_TOS_DATE

Novo na versão 4.15.

Nota: Você precisa de [Legal](#) instalado para fazer isso funcionar.

Data da última atualização dos documentos de termos de serviço. Sempre que a data muda, os usuários são obrigados a concordar com os termos de serviço.

```
from datetime import date

LEGAL_TOS_DATE = date(2022, 2, 2)
```

2.16.51 LEGAL_URL

Novo na versão 3.5.

URL onde sua instância de Weblate mostra seus documentos legais.

Dica: Útil se você hospeda seus documentos legais fora do Weblate para incorporá-los ao Weblate, verifique [Legal](#) para obter detalhes.

Exemplo:

```
LEGAL_URL = "https://weblate.org/terms/"
```

Ver também:

[PRIVACY_URL](#)

2.16.52 LICENSE_EXTRA

Licenças adicionais para incluir nas opções de licença.

Nota: Cada definição de licença deve ser uma tupla de seu nome curto, um nome longo e uma URL.

Por exemplo:

```
LICENSE_EXTRA = [
    (
        "AGPL-3.0",
        "GNU Affero General Public License v3.0",
        "https://www.gnu.org/licenses/agpl-3.0-standalone.html",
    ),
]
```

2.16.53 LICENSE_FILTER

Alterado na versão 4.3: Configurando este para valor em branco agora desabilita o alerta de licença.

Filtrar licenças da lista para mostrar. Isto também desabilita o alerta de licença quando configurado para vazio.

Nota: Este filtro usa os nomes de licença curtos.

Por exemplo:

```
LICENSE_FILTER = {"AGPL-3.0", "GPL-3.0-or-later"}
```

A seguir, desativa o alerta de licença:

```
LICENSE_FILTER = set ()
```

Ver também:

[alerts](#)

2.16.54 LICENSE_REQUIRED

Define se o atributo de licença em *Configuração de componente* é necessário.

Nota: Isso está desativado por padrão.

2.16.55 LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH

Se o comprimento de uma determinada tradução deve ser limitado. A restrição é o comprimento do texto fonte × 10 caracteres.

Dica: Defina isso como `False` para permitir traduções mais longas (até 10.000 caracteres) independentemente do comprimento do texto fonte.

Nota: O padrão é `True`.

2.16.56 LOCALIZE_CDN_URL e LOCALIZE_CDN_PATH

Essas configurações definem a extensão *CDN de localização do JavaScript*. `LOCALIZE_CDN_URL` define a URL raiz onde o CDN de localização está disponível e `LOCALIZE_CDN_PATH` define o caminho onde o Weblate deve armazenar arquivos gerados que serão servidos em `LOCALIZE_CDN_URL`.

Dica: No Hosted Weblate, é usada com `https://weblate-cdn.com/`.

Ver também:

CDN de localização do JavaScript

2.16.57 LOGIN_REQUIRED_URLS

Uma lista de URLs que você deseja exigir autenticação. (Além das regras padrão incorporadas na Weblate).

Dica: Isso permite que você proteja com senha toda uma instalação usando:

```
LOGIN_REQUIRED_URLS = (r"/(.*)$", )
REST_FRAMEWORK["DEFAULT_PERMISSION_CLASSES"] = [
    "rest_framework.permissions.IsAuthenticated"
]
```

Dica: É desejável bloquear o acesso à API também, como mostrado no exemplo acima.

Ver também:

`REQUIRE_LOGIN`

2.16.58 LOGIN_REQUIRED_URLS_EXCEPTIONS

Lista de exceções para `LOGIN_REQUIRED_URLS`. Se não especificado, os usuários podem acessar a página de autenticação.

Algumas das exceções que você pode querer incluir:

```
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r"/accounts/(.*)$", # Required for sign in
    r"/static/(.*)$", # Required for development mode
    r"/widgets/(.*)$", # Allowing public access to widgets
    r"/data/(.*)$", # Allowing public access to data exports
    r"/hooks/(.*)$", # Allowing public access to notification hooks
    r"/api/(.*)$", # Allowing access to API
    r"/js/i18n/$", # JavaScript localization
)
```

2.16.59 MATOMO_SITE_ID

ID de um site em Matomo (anteriormente Piwik) que você quer rastrear.

Nota: Essa integração não tem suporte ao Matomo Tag Manager.

Ver também:

`MATOMO_URL`

2.16.60 MATOMO_URL

URL completa (incluindo barra ao final) de uma instalação Matomo (anteriormente Piwik) que você deseja usar para rastrear o uso do Weblate. Por favor, consulte <<https://matomo.org/>> para mais detalhes.

Dica: Essa integração não tem suporte ao Matomo Tag Manager.

Por exemplo:

```
MATOMO_SITE_ID = 1
MATOMO_URL = "https://example.matomo.cloud/"
```

Ver também:

`MATOMO_SITE_ID`

2.16.61 NEARBY_MESSAGES

Quantos textos devem ser mostrados em torno do texto traduzido atualmente. Este é apenas um valor padrão, os usuários podem ajustar isso em *Perfil do usuário*.

2.16.62 DEFAULT_PAGE_LIMIT

Novo na versão 4.7.

Número padrão de elementos a serem exibidos quando a paginação está ativa.

2.16.63 PAGURE_CREDENTIALS

Novo na versão 4.3.2.

Lista para credenciais para servidores Pagure.

```
PAGURE_CREDENTIALS = {
    "pagure.io": {
        "username": "weblate",
        "token": "your-api-token",
    },
    "pagure.example.com": {
        "username": "weblate",
        "token": "another-api-token",
    },
}
```

Ver também:

Merge requests do Pagure, *Pagure API*

2.16.64 PRIVACY_URL

Novo na versão 4.8.1.

URL onde sua instância de Weblate mostra sua política de privacidade.

Dica: Útil se você hospeda seus documentos legais fora do Weblate para incorporá-los ao Weblate, verifique [Legal](#) para obter detalhes.

Exemplo:

```
PRIVACY_URL = "https://weblate.org/terms/"
```

Ver também:

LEGAL_URL

2.16.65 PRIVATE_COMMIT_EMAIL_OPT_IN

Novo na versão 4.15.

Configura se o e-mail de commit privado foi optado ou não (por padrão é não optado).

Ver também:

Perfil, *PRIVATE_COMMIT_EMAIL_TEMPLATE*

2.16.66 PRIVATE_COMMIT_EMAIL_TEMPLATE

Novo na versão 4.15.

Modelo para gerar e-mail de commit privado para um usuário. O padrão é "{username}@users.noreply.{site_domain}".

Defina para texto vazio para desabilitar.

Nota: O uso de e-mail de commit diferente é opcional para os usuários, a menos que configurado por `PRIVATE_COMMIT_EMAIL_OPT_IN`. Os usuários podem configurar o e-mail de commit no *Perfil*.

2.16.67 PROJECT_BACKUP_KEEP_COUNT

Novo na versão 4.14.

Define quantos backups por projeto são mantidos no servidor. A configuração padrão é 3.

Ver também:

Backups de nível de projeto

2.16.68 PROJECT_BACKUP_KEEP_DAYS

Novo na versão 4.14.

Define por quanto tempo os backups do projeto serão mantidos no servidor. O tempo padrão é de 30 dias.

Ver também:

Backups de nível de projeto

2.16.69 PROJECT_NAME_RESTRICT_RE

Novo na versão 4.15.

Define uma expressão regular para restringir a nomenclatura do projeto. Quaisquer nomes correspondentes serão rejeitados.

Ver também:

Nome do projeto

2.16.70 PROJECT_WEB_RESTRICT_RE

Novo na versão 4.15.

Define uma expressão regular para restringir os sites do projeto. Quaisquer URLs correspondentes serão rejeitados.

Ver também:

Site do projeto

2.16.71 RATELIMIT_ATTEMPTS

Novo na versão 3.2.

O número máximo de tentativas de autenticação antes da limitação da taxa ser aplicada.

O padrão é 5.

Ver também:

Limitação de taxa, `RATELIMIT_WINDOW`, `RATELIMIT_LOCKOUT`

2.16.72 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 tendo como padrão 300 (5 minutos).

Ver 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 tendo como padrão 600 (10 minutos).

Ver 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 registro. Isso só limita novos registros, os usuários ainda podem autenticar e adicionar autenticação usando todos os backends de autenticação configurados.

É recomendado para manter `REGISTRATION_OPEN` habilitado enquanto limita os backends de registro, caso contrário, os usuários poderão se registrar, mas o Weblate não mostrará links para se registrar na interface do usuário.

Exemplo:

```
REGISTRATION_ALLOW_BACKENDS = ["azuread-oauth2", "azuread-tenant-oauth2"]
```

Dica: Os nomes de backend correspondem aos nomes usados na URL para autenticação.

Ver também:

`REGISTRATION_OPEN`, *Autenticação*

2.16.75 REGISTRATION_CAPTCHA

Um valor de `True` ou `False` indicando se o registro de novas contas é protegido pelo CAPTCHA. Esta configuração é opcional, e um padrão de `True` será presumido se não for fornecido.

Se ativado, um CAPTCHA é adicionado a todas as páginas onde um usuário digita seu endereço de e-mail:

- Registro de nova conta.
- Recuperação de senha.
- Adição de e-mail a uma conta.
- Formulário de contato para usuários que não estão autenticados.

2.16.76 REGISTRATION_EMAIL_MATCH

Novo na versão 2.17.

Permite filtrar quais endereços de e-mail podem ser registrados.

O padrão é `.*`, que permite que qualquer endereço de e-mail seja registrado.

Você pode usá-lo para restringir o registro a um único domínio de e-mail:

```
REGISTRATION_EMAIL_MATCH = r"^.*@weblate\.org$"
```

2.16.77 REGISTRATION_OPEN

Se o registro de novas contas é atualmente permitido. Essa configuração opcional pode permanecer com o padrão `True`, ou pode ser alterada para `Falsa`.

Essa configuração afeta a autenticação embutida por endereço de e-mail ou através do Python Social Auth (você pode adicionar à lista de permissão certos back-ends usando `REGISTRATION_ALLOW_BACKENDS`).

Nota: Se estiver usando métodos de autenticação de terceiros, como *Autenticação por LDAP*, ele apenas oculta o formulário de registro, mas novos usuários ainda conseguem se autenticar e criar contas.

Ver também:

`REGISTRATION_ALLOW_BACKENDS`, `REGISTRATION_EMAIL_MATCH`, *Autenticação*

2.16.78 REGISTRATION_REBIND

Novo na versão 4.16.

Allow rebinding authentication backends for existing users. Turn this on when migrating between authentication providers.

Nota: Disabled by default to not allow adding other authentication backends to existing account. Rebinding can lead to account compromise when using more third-party authentication backends.

2.16.79 REPOSITORY_ALERT_THRESHOLD

Novo na versão 4.0.2.

Limiar para acionar um alerta para repositórios desatualizados ou aqueles que contenham muitas alterações. O padrão é 25.

Ver também:

alerts

2.16.80 REQUER_LOGIN

Novo na versão 4.1.

Isso habilita `LOGIN_REQUIRED_URLS` 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`.

2.16.81 SENTRY_DSN

Novo na versão 3.9.

DSN do Sentry para usar para *Coletando relatórios de erros*.

Ver também:

Integração Django para o Sentry

2.16.82 IDADE_REGISTRO_SESSÃO_AUTENTICADO

Novo na versão 4.3.

Configura a expiração da sessão para usuário autenticados. Isso complementa `django:IDADE_REGISTRO_SESSÃO` que é utilizado por usuários não autenticados.

Ver também:

`SESSION_COOKIE_AGE`

2.16.83 SIMPLIFY_LANGUAGES

Usa códigos de idioma simples para combinações padrão de idioma/país. Por exemplo, uma tradução de `fr_FR` usará o código de idioma `fr`. Este é geralmente o comportamento desejado, pois simplifica a lista de idiomas para essas combinações padrão.

Desative isso se quiser traduções diferentes para cada variante.

2.16.84 SITE_DOMAIN

Configura o domínio do site. Isso é necessário para produzir links absolutos corretos em muitos escopos (por exemplo, ativação de e-mails, notificações ou feeds RSS).

No caso de o Weblate estar sendo executado em um porte fora do padrão, inclua-a aqui também.

Exemplos:

```
# Production site with domain name
SITE_DOMAIN = "weblate.example.com"

# Local development with IP address and port
SITE_DOMAIN = "127.0.0.1:8000"
```

Nota: Esta configuração deve conter apenas o nome de domínio. Para configurar o protocolo (habilitar e aplicar HTTPS), use `ENABLE_HTTPS` e for alterar URL, use `URL_PREFIX`.

Dica: Em um contêiner Docker, o domínio do site é configurado através de `WEBLATE_ALLOWED_HOSTS`.

Ver também:

Definir domínio correto do site, Configuração de hosts permitidos, Configurar corretamente HTTPS WE-BLATE_SITE_DOMAIN, ENABLE_HTTPS

2.16.85 SITE_TITLE

Título do site a ser usado para o site e e-mails enviados.

2.16.86 SPECIAL_CHARS

Caracteres adicionais para incluir no teclado visual, *Teclado visual*.

O valor padrão é:

```
SPECIAL_CHARS = ("\t", "\n", "\u00a0", "...")
```

2.16.87 SINGLE_PROJECT

Novo na versão 3.8.

Redireciona os usuários diretamente para um projeto ou componente em vez de mostrar o painel. Você pode configurá-lo como `True` e, neste caso, ele só funciona no caso de haver realmente apenas um único projeto no Weblate. Alternativamente, defina o projeto, e ele redirecionará incondicionalmente para este projeto.

Alterado na versão 3.11: A configuração agora também aceita um slug de projeto, para forçar a exibição desse único projeto.

Exemplo:

```
SINGLE_PROJECT = "test"
```

2.16.88 SSH_EXTRA_ARGS

Novo na versão 4.9.

Permite adicionar parâmetros personalizados quando o Weblate está chamando SSH. Isso é útil ao se conectar a servidores usando 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*, você pode habilitar isso usando:

```
SSH_EXTRA_ARGS = "-oKexAlgorithms=+diffie-hellman-group1-sha1"
```

Dica: O texto é avaliado pelo shell, portanto, certifique-se de colocar aspas em todos os espaços em branco e caracteres especiais.

Ver também:

[Opções legadas do OpenSSH](#)

2.16.89 STATUS_URL

A URL onde sua instância de Weblate relata seu status.

2.16.90 SUGGESTION_CLEANUP_DAYS

Novo na versão 3.2.1.

Exclui automaticamente sugestões após um determinado número de dias. O padrão é `None`, ou seja, sem exclusões.

2.16.91 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.

Ver também:

[Definições embutidas de idioma](#)

2.16.92 URL_PREFIX

Esta configuração permite que você execute Weblate em algum caminho (caso contrário, ele depende de ser executado a partir da raiz do servidor web).

Nota: Para usar esta configuração, você também precisa configurar seu servidor para remover este prefixo. Por exemplo, com o WSGI, isso pode ser alcançado definindo `WSGIScriptAlias`.

Dica: O prefixo deve iniciar com um `/`.

Exemplo:

```
URL_PREFIX = "/translations"
```

Nota: Esta configuração não funciona com o servidor embutido do Django, você teria que ajustar `urls.py` para conter este prefixo.

2.16.93 VCS_API_DELAY

Novo na versão 4.15.1.

Configura atraso mínimo em segundos entre chamadas de API de terceiros em *Pull requests do GitHub*, *Merge requests do GitLab*, *Pull requests do Gitea*, and *Merge requests do Pagure*.

Limitador de chamadas de API do Weblate para esses serviços para evitar sobrecarregá-los.

Se você estiver sendo afetado pelo limitador de taxa secundário no GitHub, aumentar isso pode ajudar.

O valor padrão é 10.

2.16.94 VCS_BACKENDS

Configuração de backends VCS disponíveis.

Nota: Weblate tenta usar todos os back-ends suportados para os seus usuários.

Dica: Você pode limitar escolhas ou adicionar back-ends VCS personalizados usando isso.

```
VCS_BACKENDS = ("weblate.vcs.git.GitRepository",)
```

Ver também:

Integração com controle de versão

2.16.95 VCS_CLONE_DEPTH

Novo na versão 3.10.2.

Configura o quão profunda a clonagem de repositórios Weblate deve ir.

Nota: Atualmente, isso só é suportado em *Git*. Por padrão, o Weblate faz clones rasos dos repositórios para tornar a clonagem mais rápida e economizar espaço em disco. Dependendo do seu uso (por exemplo, ao usar o personalizado *Extensões*), você pode querer aumentar a profundidade ou desligar os clones rasos completamente definindo isso para 0.

Dica: No caso de você receber erro fatal: `protocol error: expected old/new/ref, got 'shallow <hash de commit>'` ao fazer push do Weblate, desative clones rasos completamente configurando:

```
VCS_CLONE_DEPTH = 0
```

2.16.96 WEBLATE_ADDONS

Lista de extensões disponíveis para uso. Para usá-las, elas devem ser habilitadas para um determinado componente de tradução. Por padrão, isso inclui todas as extensões embutidas, ao estender a lista, você provavelmente vai querer manter as existentes habilitadas, por exemplo:

```
WEBLATE_ADDONS = (  
    # Built-in 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.xml.XMLCustomizeAddon",  
    "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.

Ver também:

Extensões, DEFAULT_ADDONS

2.16.97 WEBLATE_EXPORTERS

Novo na versão 4.2.

Lista de exportadores disponíveis que oferecem download de traduções ou glossários em vários formatos de arquivo.

Ver também:

Formatos de arquivos suportados

2.16.98 WEBLATE_FORMATS

Novo na versão 3.0.

Lista de formatos de arquivo disponíveis para uso.

Nota: A lista padrão já tem os formatos comuns.

Ver também:

Formatos de arquivos suportados

2.16.99 WEBLATE_MACHINERY

Novo na versão 4.13.

Lista de serviços de tradução disponíveis para uso.

Ver também:

Configurando sugestões automáticas

2.16.100 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.

Ver também:

Assinando commits do Git com GnuPG

2.16.101 WEBSITE_REQUIRED

Define se *Site 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 © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

import os
import platform
from logging.handlers import SysLogHandler

# Title of site to use
SITE_TITLE = "Weblate"
```

(continua na próxima página)

(continuação da página anterior)

```

# 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:
            # "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"
CACHE_DIR = f"{DATA_DIR}/cache"

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name

```

(continua na próxima página)

(continuação da página anterior)

```
# although not all choices may be available on all operating systems.
# In a Windows environment this must be set to your system time zone.
TIME_ZONE = "UTC"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

LANGUAGES = (
    ("ar", "العربية"),
    ("az", "Azərbaycan"),
    ("be", "Беларуская"),
    ("be@latin", "Biełaruskaja"),
    ("bg", "Български"),
    ("br", "Brezhoneg"),
    ("ca", "Català"),
    ("cs", "Čeština"),
    ("da", "Dansk"),
    ("de", "Deutsch"),
    ("en", "English"),
    ("el", "Ελληνικά"),
    ("en-gb", "English (United Kingdom)"),
    ("es", "Español"),
    ("fi", "Suomi"),
    ("fr", "Français"),
    ("gl", "Galego"),
    ("he", "עברית"),
    ("hu", "Magyar"),
    ("hr", "Hrvatski"),
    ("id", "Indonesia"),
    ("is", "Íslenska"),
    ("it", "Italiano"),
    ("ja", "日本語"),
    ("kab", "Taqbaylit"),
    ("kk", "Қазақ тілі"),
    ("ko", "한국어"),
    ("nb", "Norsk bokmål"),
    ("nl", "Nederlands"),
    ("pl", "Polski"),
    ("pt", "Português"),
    ("pt-br", "Português brasileiro"),
    ("ro", "Română"),
    ("ru", "Русский"),
    ("sk", "Slovenčina"),
    ("sl", "Slovenščina"),
    ("sq", "Shqip"),
    ("sr", "Српски"),
    ("sr-latn", "Srpski"),
    ("sv", "Svenska"),
    ("th", "ไทย"),
    ("tr", "Türkçe"),
    ("uk", "Українська"),
    ("zh-hans", "简体中文"),
    ("zh-hant", "繁體中文"),
)

SITE_ID = 1

# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True
```

(continua na próxima página)

(continuação da página anterior)

```

# 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
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",
            ],

```

(continua na próxima página)

(continuação da página anterior)

```

    },
    "APP_DIRS": True,
}
]

# GitHub username and token for sending pull requests.
# Please see the documentation for more details.
GITHUB_CREDENTIALS = {}

# GitLab username and token for sending merge requests.
# Please see the documentation for more details.
GITLAB_CREDENTIALS = {}

# Bitbucket username and token for sending merge requests.
# Please see the documentation for more details.
BITBUCKETSERVER_CREDENTIALS = {}

# Authentication configuration
AUTHENTICATION_BACKENDS = (
    "social_core.backends.email.EmailAuth",
    # "social_core.backends.google.GoogleOAuth2",
    # "social_core.backends.github.GithubOAuth2",
    # "social_core.backends.bitbucket.BitbucketOAuth2",
    # "social_core.backends.suse.OpenSUSEOpenId",
    # "social_core.backends.ubuntu.UbuntuOpenId",
    # "social_core.backends.fedora.FedoraOpenId",
    # "social_core.backends.facebook.FacebookOAuth2",
    "weblate.accounts.auth.WeblateUserBackend",
)

# Custom user model
AUTH_USER_MODEL = "weblate_auth.User"

# Social auth backends setup
SOCIAL_AUTH_GITHUB_KEY = ""
SOCIAL_AUTH_GITHUB_SECRET = ""
SOCIAL_AUTH_GITHUB_SCOPE = ["user:email"]

SOCIAL_AUTH_GITHUB_ORG_KEY = ""
SOCIAL_AUTH_GITHUB_ORG_SECRET = ""
SOCIAL_AUTH_GITHUB_ORG_NAME = ""

SOCIAL_AUTH_GITHUB_TEAM_KEY = ""
SOCIAL_AUTH_GITHUB_TEAM_SECRET = ""
SOCIAL_AUTH_GITHUB_TEAM_ID = ""

SOCIAL_AUTH_BITBUCKET_OAUTH2_KEY = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_SECRET = ""
SOCIAL_AUTH_BITBUCKET_OAUTH2_VERIFIED_EMAILS_ONLY = True

SOCIAL_AUTH_FACEBOOK_KEY = ""
SOCIAL_AUTH_FACEBOOK_SECRET = ""
SOCIAL_AUTH_FACEBOOK_SCOPE = ["email", "public_profile"]
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {"fields": "id,name,email"}

SOCIAL_AUTH_GOOGLE_OAUTH2_KEY = ""
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET = ""

# Social auth settings
SOCIAL_AUTH_PIPELINE = (

```

(continua na próxima página)

(continuação da página anterior)

```

    "social_core.pipeline.social_auth.social_details",
    "social_core.pipeline.social_auth.social_uid",
    "social_core.pipeline.social_auth.auth_allowed",
    "social_core.pipeline.social_auth.social_user",
    "weblate.accounts.pipeline.store_params",
    "weblate.accounts.pipeline.verify_open",
    "social_core.pipeline.user.get_username",
    "weblate.accounts.pipeline.require_email",
    "social_core.pipeline.mail.mail_validation",
    "weblate.accounts.pipeline.revoke_mail_code",
    "weblate.accounts.pipeline.ensure_valid",
    "weblate.accounts.pipeline.remove_account",
    "social_core.pipeline.social_auth.associate_by_email",
    "weblate.accounts.pipeline.reauthenticate",
    "weblate.accounts.pipeline.verify_username",
    "social_core.pipeline.user.create_user",
    "social_core.pipeline.social_auth.associate_user",
    "social_core.pipeline.social_auth.load_extra_data",
    "weblate.accounts.pipeline.cleanup_next",
    "weblate.accounts.pipeline.user_full_name",
    "weblate.accounts.pipeline.store_email",
    "weblate.accounts.pipeline.notify_connect",
    "weblate.accounts.pipeline.password_reset",
)
SOCIAL_AUTH_DISCONNECT_PIPELINE = (
    "social_core.pipeline.disconnect.allowed_to_disconnect",
    "social_core.pipeline.disconnect.get_entries",
    "social_core.pipeline.disconnect.revoke_tokens",
    "weblate.accounts.pipeline.cycle_session",
    "weblate.accounts.pipeline.adjust_primary_mail",
    "weblate.accounts.pipeline.notify_disconnect",
    "social_core.pipeline.disconnect.disconnect",
    "weblate.accounts.pipeline.cleanup_next",
)

# Custom authentication strategy
SOCIAL_AUTH_STRATEGY = "weblate.accounts.strategy.WeblateStrategy"

# Raise exceptions so that we can handle them later
SOCIAL_AUTH_RAISE_EXCEPTIONS = True

SOCIAL_AUTH_EMAIL_VALIDATION_FUNCTION = "weblate.accounts.pipeline.send_validation"
SOCIAL_AUTH_EMAIL_VALIDATION_URL = f"{URL_PREFIX}/accounts/email-sent/"
SOCIAL_AUTH_LOGIN_ERROR_URL = f"{URL_PREFIX}/accounts/login/"
SOCIAL_AUTH_EMAIL_FORM_URL = f"{URL_PREFIX}/accounts/email/"
SOCIAL_AUTH_NEW_ASSOCIATION_REDIRECT_URL = f"{URL_PREFIX}/accounts/profile/#account
↪"
SOCIAL_AUTH_PROTECTED_USER_FIELDS = ("email",)
SOCIAL_AUTH_SLUGIFY_USERNAMES = True
SOCIAL_AUTH_SLUGIFY_FUNCTION = "weblate.accounts.pipeline.slugify_username"

# Password validation configuration
AUTH_PASSWORD_VALIDATORS = [
    {
        "NAME": "django.contrib.auth.password_validation.
↪UserAttributeSimilarityValidator" # noqa: E501, pylint: disable=line-too-long
    },
    {
        "NAME": "django.contrib.auth.password_validation.MinimumLengthValidator",
        "OPTIONS": {"min_length": 10},
    },
]

```

(continua na próxima página)

(continuação da página anterior)

```

    {"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",
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "weblate.accounts.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
    "social_django.middleware.SocialAuthExceptionMiddleware",
    "weblate.accounts.middleware.RequireLoginMiddleware",
    "weblate.api.middleware.ThrottlingMiddleware",
    "weblate.middleware.SecurityMiddleware",
    "weblate.wladmin.middleware.ManageMiddleware",
]

ROOT_URLCONF = "weblate.urls"

# Django and Weblate apps
INSTALLED_APPS = [
    # Weblate apps on top to override Django locales and templates
    "weblate.addons",
    "weblate.auth",
    "weblate.checks",
    "weblate.formats",
    "weblate.glossary",
    "weblate.machinery",
    "weblate.trans",
    "weblate.lang",
    "weblate_language_data",
    "weblate.memory",

```

(continua na próxima página)

(continuação da página anterior)

```

    "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",
    "crispy_bootstrap3",
    "compressor",
    "rest_framework",
    "rest_framework.authtoken",
    "django_filters",
    "django_celery_beat",
    "corsheaders",
]

# Custom exception reporter to include some details
DEFAULT_EXCEPTION_REPORTER_FILTER = "weblate.trans.debug.
↳ WeblateExceptionReporterFilter"

# Default logging of Weblate messages
# - to syslog in production (if available)
# - otherwise to console
# - you can also choose "logfile" to log into separate file
#   after configuring it below

# Detect if we can connect to syslog
HAVE_SYSLOG = False
if platform.system() != "Windows":
    try:
        handler = SysLogHandler(address="/dev/log", facility=SysLogHandler.LOG_
↳ LOCAL2)
        handler.close()
        HAVE_SYSLOG = True
    except OSError:
        HAVE_SYSLOG = False

DEFAULT_LOG = "console" if DEBUG or not HAVE_SYSLOG else "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

```

(continua na próxima página)

(continuação da página anterior)

```
# more details on how to customize your logging configuration.
LOGGING = {
    "version": 1,
    "disable_existing_loggers": True,
    "filters": {"require_debug_false": {"()": "django.utils.log.RequireDebugFalse"}
    ↪},
    "formatters": {
        "syslog": {"format": "weblate[%(process)d]: %(levelname)s %(message)s"},
        "simple": {"format": "[% (asctime)s: %(levelname)s/%(process)s] %(message)s
    ↪"},
        "logfile": {"format": "%(asctime)s %(levelname)s %(message)s"},
        "django.server": {
            "()": "django.utils.log.ServerFormatter",
            "format": "[% (server_time)s] %(message)s",
        },
    },
    "handlers": {
        "mail_admins": {
            "level": "ERROR",
            "filters": ["require_debug_false"],
            "class": "django.utils.log.AdminEmailHandler",
            "include_html": True,
        },
        "console": {
            "level": "DEBUG",
            "class": "logging.StreamHandler",
            "formatter": "simple",
        },
        "django.server": {
            "level": "INFO",
            "class": "logging.StreamHandler",
            "formatter": "django.server",
        },
        "syslog": {
            "level": "DEBUG",
            "class": "logging.handlers.SysLogHandler",
            "formatter": "syslog",
            "address": "/dev/log",
            "facility": SysLogHandler.LOG_LOCAL2,
        },
        # Logging to a file
        # "logfile": {
        #     "level": "DEBUG",
        #     "class": "logging.handlers.RotatingFileHandler",
        #     "filename": "/var/log/weblate/weblate.log",
        #     "maxBytes": 100000,
        #     "backupCount": 3,
        #     "formatter": "logfile",
        # },
    },
    "loggers": {
        "django.request": {
            "handlers": ["mail_admins", DEFAULT_LOG],
            "level": "ERROR",
            "propagate": True,
        },
        "django.server": {
            "handlers": ["django.server"],
            "level": "INFO",
            "propagate": False,
        },
    },
}
```

(continua na próxima página)

```

    # Logging database queries
    # "django.db.backends": {
    #     "handlers": [DEFAULT_LOG],
    #     "level": "DEBUG",
    # },
    "weblate": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Logging VCS operations
    "weblate.vcs": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Python Social Auth
    "social": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # Django Authentication Using LDAP
    "django_auth_ldap": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    # SAML IdP
    "djangosaml2idp": {"handlers": [DEFAULT_LOG], "level": DEFAULT_LOGLEVEL},
    },
}

# Remove syslog setup if it's not present
if not HAVE_SYSLOG:
    del LOGGING["handlers"]["syslog"]

# List of machine translations
MT_SERVICES = (
    # "weblate.machinery.apertium.ApertiumAPYTranslation",
    # "weblate.machinery.baidu.BaiduTranslation",
    # "weblate.machinery.deepl.DeepLTranslation",
    # "weblate.machinery.glosbe.GlosbeTranslation",
    # "weblate.machinery.google.GoogleTranslation",
    # "weblate.machinery.googlev3.GoogleV3Translation",
    # "weblate.machinery.libretranslate.LibreTranslateTranslation",
    # "weblate.machinery.microsoft.MicrosoftCognitiveTranslation",
    # "weblate.machinery.microsoftterminology.MicrosoftTerminologyService",
    # "weblate.machinery.modernmt.ModernMTTranslation",
    # "weblate.machinery.mymemory.MyMemoryTranslation",
    # "weblate.machinery.netease.NeteaseSightTranslation",
    # "weblate.machinery.tmserver.AmagamaTranslation",
    # "weblate.machinery.tmserver.TMServerTranslation",
    # "weblate.machinery.yandex.YandexTranslation",
    # "weblate.machinery.saptranslationhub.SAPTranslationHub",
    # "weblate.machinery.youdao.YoudaoTranslation",
    "weblate.machinery.weblatetm.WeblateTranslation",
    "weblate.memory.machine.WeblateMemory",
)

# Machine translation API keys

# URL of the Apertium APY server
MT_APERTIUM_APY = None

# DeepL API key
MT_DEEPL_KEY = None

# LibreTranslate
MT_LIBRETRANSLATE_API_URL = None
MT_LIBRETRANSLATE_KEY = None

# Microsoft Cognitive Services Translator API, register at
# https://portal.azure.com/
MT_MICROSOFT_COGNITIVE_KEY = None
MT_MICROSOFT_REGION = None

```


(continuação da página anterior)

```

# 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

# 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

```

(continua na próxima página)

(continuação da página anterior)

```

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
# Allow more fields for case with a lot of subscriptions in profile
DATA_UPLOAD_MAX_NUMBER_FIELDS = 2000

# Apply session coookie settings to language cookie as ewll
LANGUAGE_COOKIE_SECURE = SESSION_COOKIE_SECURE
LANGUAGE_COOKIE_HTTPONLY = SESSION_COOKIE_HTTPONLY
LANGUAGE_COOKIE_AGE = SESSION_COOKIE_AGE_AUTHENTICATED * 10
LANGUAGE_COOKIE_SAMESITE = SESSION_COOKIE_SAMESITE

# Some security headers
SECURE_BROWSER_XSS_FILTER = True
X_FRAME_OPTIONS = "DENY"
SECURE_CONTENT_TYPE_NOSNIFF = True

# Optionally enable HSTS
SECURE_HSTS_SECONDS = 31536000 if ENABLE_HTTPS else 0
SECURE_HSTS_PRELOAD = ENABLE_HTTPS
SECURE_HSTS_INCLUDE_SUBDOMAINS = ENABLE_HTTPS

# HTTPS detection behind reverse proxy
SECURE_PROXY_SSL_HEADER = None

# URL of login
LOGIN_URL = f"{URL_PREFIX}/accounts/login/"

# URL of logout
LOGOUT_URL = f"{URL_PREFIX}/accounts/logout/"

# Default location for login
LOGIN_REDIRECT_URL = f"{URL_PREFIX}/"

# Anonymous user name
ANONYMOUS_USER_NAME = "anonymous"

# Reverse proxy settings
IP_PROXY_HEADER = "HTTP_X_FORWARDED_FOR"
IP_BEHIND_REVERSE_PROXY = False
IP_PROXY_OFFSET = 0

# Sending HTML in mails
EMAIL_SEND_HTML = True

# Subject of emails includes site title
EMAIL_SUBJECT_PREFIX = f"[{SITE_TITLE}] "

# Enable remote hooks
ENABLE_HOOKS = True

# 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

```

(continua na próxima página)

(continuação da página anterior)

```

# 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_ALLOWED_TEMPLATE_PACKS = "bootstrap3"
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",
#     "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",

```

(continua na próxima página)

(continuação da página anterior)

```

# "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.xml.XMLCustomizeAddon",
#     "weblate.addons.properties.PropertiesSortAddon",
#     "weblate.addons.git.GitSquashAddon",
#     "weblate.addons.removal.RemoveComments",
#     "weblate.addons.removal.RemoveSuggestions",
#     "weblate.addons.resx.ResxUpdateAddon",
#     "weblate.addons.yaml.YAMLCustomizeAddon",
#     "weblate.addons.cdn.CDNJSAddon",
# )

# E-mail address that error messages come from.
SERVER_EMAIL = "noreply@example.com"

# Default email address to use for various automated correspondence from
# the site managers. Used for registration emails.
DEFAULT_FROM_EMAIL = "noreply@example.com"

```

(continua na próxima página)

(continuação da página anterior)

```

# 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",
        "TIMEOUT": 3600,
    },
    "avatar": {
        "BACKEND": "django.core.cache.backends.filebased.FileBasedCache",
        "LOCATION": os.path.join(DATA_DIR, "avatar-cache"),
        "TIMEOUT": 86400,
        "OPTIONS": {"MAX_ENTRIES": 1000},
    },
}

# Store sessions in cache
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# Store messages in session
MESSAGE_STORAGE = "django.contrib.messages.storage.session.SessionStorage"

# REST framework settings for API
REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth` permissions,
    # or allow read-only access for unauthenticated users.
    "DEFAULT_PERMISSION_CLASSES": [
        # Require authentication for login required sites
        "rest_framework.permissions.IsAuthenticated"
        if REQUIRE_LOGIN
        else "rest_framework.permissions.IsAuthenticatedOrReadOnly"
    ],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework.authentication.TokenAuthentication",
        "weblate.api.authentication.BearerAuthentication",
        "rest_framework.authentication.SessionAuthentication",
    ),
    "DEFAULT_THROTTLE_CLASSES": (
        "weblate.api.throttling.UserRateThrottle",
        "weblate.api.throttling.AnonRateThrottle",
    ),
    "DEFAULT_THROTTLE_RATES": {"anon": "100/day", "user": "5000/hour"},
    "DEFAULT_PAGINATION_CLASS": "weblate.api.pagination.StandardPagination",
    "PAGE_SIZE": 50,
    "VIEW_DESCRIPTION_FUNCTION": "weblate.api.views.get_view_description",
    "UNAUTHENTICATED_USER": "weblate.auth.models.get_anonymous",
}

# Fonts CDN URL

```

(continua na próxima página)

(continuação da página anterior)

```

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_SCHEDULER = "django_celery_beat.schedulers:DatabaseScheduler"
CELERY_TASK_ROUTES = {
    "weblate.trans.tasks.auto_translate*": {"queue": "translate"},
    "weblate.accounts.tasks.notify_*": {"queue": "notify"},
    "weblate.accounts.tasks.send_mails": {"queue": "notify"},
    "weblate.utils.tasks.settings_backup": {"queue": "backup"},
    "weblate.utils.tasks.database_backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup": {"queue": "backup"},
    "weblate.wladmin.tasks.backup_service": {"queue": "backup"},
    "weblate.memory.tasks.*": {"queue": "memory"},
}

```

(continua na próxima página)

(continuação da página anterior)

```
# CORS allowed origins
CORS_ALLOWED_ORIGINS = []
CORS_URLS_REGEX = r"^/api/.*$"

# 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 gerenciamento sob um usuário diferente daquele que executa seu servidor web pode resultar em arquivos recebendo permissões erradas, por favor, confira [Permissões do sistema de arquivos](#) para mais detalhes.

Você encontrará comandos básicos de gerenciamento (disponíveis como `./manage.py` nas fontes do Django ou como um conjunto estendido em um script chamado **weblate** instalável em cima do Weblate).

2.18.1 Invocando comandos de gerenciamento

Como mencionado anteriormente, sua invocação depende de como você instalou o Weblate.

Se estiver usando o virtualenv para Weblate, você 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 você estiver usando o código-fonte diretamente (de um tarball ou checkout Git), o script de gerenciamento é `./manage.py` disponível nos fontes do Weblate. Para executá-lo:

```
python ./manage.py list_versions
```

Se você instalou o Weblate usando o instalador pip, ou usando o script `./setup.py`, o **weblate** é instalado em seu caminho (ou caminho virtualenv), de onde você pode usá-lo para controlar o Weblate:

```
weblate list_versions
```

Para a imagem Docker, o script é instalado como acima e você pode executá-lo usando **docker exec**:

```
docker exec --user weblate <container> weblate list_versions
```

Para **docker-compose** o processo é semelhante, você apenas tem que usar **docker-compose exec**:

```
docker-compose exec --user weblate weblate weblate list_versions
```

Caso precise passar um arquivo para ele, você pode adicionar temporariamente um volume:

```
docker-compose exec --user weblate /tmp:/tmp weblate weblate importusers /tmp/  
↪users.json
```

Ver também:

Instalando usando Docker, Instalando no Debian e Ubuntu, Instalando no SUSE e openSUSE, Instalando no RedHat, Fedora e CentOS, Instalando a partir do código-fonte

2.18.2 add_suggestions

weblate add_suggestions <project> <component> <language> <file>

Novo na versão 2.5.

Importa uma tradução do arquivo para usar como sugestão para a tradução fornecida. Ele pula traduções duplicadas; apenas os diferentes são adicionados.

--author USER@EXAMPLE.COM

E-mail do autor para sugestões. Este usuário 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 usuário listado como autor das traduções. “Usuário 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, o padrão 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
```

Ver 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.

Você 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.

You can either define which project or component to update (for example `weblate/application`), or use `--all` to update all existing components, or use `--file-format` to filter based on the file format.

2.18.7 commit_pending

weblate commit_pending <project|project/component>

Faz commit de alterações pendentes anteriores a uma determinada idade.

Você 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*.

Ver também:

Executando 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.

Ver também:

Executando 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.

Ver também:

Repositórios SSH

2.18.10 createadmin

weblate createadmin

Cria uma conta `admin` com uma senha aleatória, a menos que seja especificada.

--password PASSWORD

Fornece uma senha na linha de comando, para não gerar uma senha aleatória.

--no-password

Não define a senha, 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 usuário existente (você pode usar isso para alterar as senhas).

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 arquivo JSON contendo conteúdo de memória de tradução do Weblate.

Ver também:

Memória de tradução, Esquema de memória de tradução do Weblate

2.18.12 dumpuserdata

weblate dumpuserdata <file.json>

Despeja dados de usuário em um arquivo para uso posterior por *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.

Cria um projeto de demonstração com componentes baseados em <https://github.com/WeblateOrg/demo>. Certifique-se de que as tarefas de Celery estejam em execução antes de executar este comando.

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 arquivo JSON importado corresponde basicamente ao objeto do componente (consulte *GET /api/components/(string:project)/(string:component)/*). Você 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 arquivo JSON:

```
[
  {
    "slug": "po",
    "name": "Gettext PO",
    "file_format": "po",
    "filemask": "po/*.po",
    "new_lang": "none"
  },
  {
    "name": "Android",
    "filemask": "android/values-*/strings.xml",
    "template": "android/values/strings.xml",
    "repo": "weblate://test/test",
    "file_format": "aresource"
  }
]
```

Ver também:

import_memory

2.18.15 import_memory

weblate import_memory <file>

Novo na versão 2.20.

Importa um arquivo 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 arquivo TMX não correspondam ao que você usa no Weblate.

Ver também:

Memória de tradução, Esquema de memória de tradução do 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 componente*, 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 arquivo.

`<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>` para o `<gitrepo>`.

O `<filemask>` define a descoberta de arquivos 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>[^\s]) / (?P<component>[^\s/]*)\.po`

A importação corresponde aos componentes existentes com base em arquivos 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 arquivo base para traduções monolíngues.

Por exemplo: `{{ component }}/res/values/string.xml`

--new-base-template TEMPLATE

Personaliza o arquivo base para adicionar novas traduções.

Por exemplo: `{{ component }}/ts/en.ts`

--file-format FORMAT

Você também pode especificar o formato de arquivo a ser usado (veja *Formatos de arquivos suportados*), o padrão é a detecção automática.

--language-regex REGEX

Você pode especificar a filtragem de idioma (veja *Configuração de componente*) com este parâmetro. Tem que ser uma expressão regular válida.

--main-component

Você 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 pode ser encontrada.

--vcs NAME

Caso você precise especificar qual sistema de controle de versão usar, você pode fazê-lo aqui. O controle de versão padrão 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 arquivo precisa ser especificado, junto com o modelo de arquivo base, e como todos os componentes e traduções estão localizados em uma única pasta:

```
weblate import_project \
  --file-format=properties \
  --base-file-template=web-app/tgol-web-app/src/main/resources/i18n/%s-I18N.
↪properties \
  tanaguru \
  https://github.com/Tanaguru/Tanaguru \
  master \
  web-app/tgol-web-app/src/main/resources/i18n/**-I18N*.properties
```

Um exemplo mais complexo de análise de nomes de arquivo para obter o componente correto e o idioma de um nome de arquivo 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 em um 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 arquivos:

```
$ 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 arquivos 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'
```

Ver também:

Exemplos mais detalhados podem ser encontrados no capítulo [starting](#), alternativamente você pode querer usar *import_json*.

2.18.17 importuserdata

weblate importuserdata <file.json>

Importa dados do usuário de um arquivo criado por *dumpuserdata*

2.18.18 importusers

weblate importusers --check <file.json>

Importa usuários do despejo JSON do banco de dados `auth_users` do Django.

--check

Com esta opção irá apenas verificar se determinado arquivo pode ser importado e reportar possíveis conflitos decorrentes de nomes de usuários ou e-mails.

Você pode descartar os usuários 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 da extensão a ser instalada. Por exemplo, `weblate.gettext.customize`.

--configuration CONFIG

Configuração codificada em JSON de uma extensão.

--update

Atualiza a configuração existente da extensão.

Você 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 saída do gettext* para todos os componentes:

```
weblate install_addon --addon weblate.gettext.customize --config '{"width": -1}' --
↪update --all
```

Ver 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 para o 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.

Você 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 suas versões.

2.18.23 loadpo

weblate loadpo <project|project/component>

Recarrega traduções de disco (por exemplo, caso você tenha feito algumas atualizações no repositório VCS).

--force

Força a atualização, mesmo que os arquivos devam estar atualizados.

--lang LANGUAGE

Limita o processamento a um único idioma.

Você pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Nota: Você raramente precisa invocar isso, o Weblate carregará automaticamente os arquivos alterados para cada atualização do VCS. Isso é necessário no caso de você 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 você queira fazer alguma manutenção no repositório subjacente.

Você pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Ver 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, você 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.

Você 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 *Push ao fazer commit* na *Configuração de componente* está ativada, que é o padrão.

2.18.27 unlock_translation

weblate unlock_translation <project|project/component>

Desbloqueia um determinado componente, disponibilizando-o para tradução.

Dica: Útil caso você queira fazer alguma manutenção no repositório subjacente.

Você pode definir qual projeto ou componente atualizar (por exemplo, `weblate/application`) ou usar `--all` para atualizar todos os componentes existentes.

Ver também:

lock_translation

2.18.28 setupgroups

weblate setupgroups

Configura grupos padrão e, opcionalmente, atribui todos os usuários a esse grupo padrão.

--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 *Controle de acesso do projeto*.

Ver também:

Lista de privilégios e funções embutidas

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.

Você 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.

Você 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 *Ganchos 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, status ou especificar metas para tradução.

Os usuários receberão notificação sobre os anúncios de projetos assistidos (a menos que optem por não participar).

Isso pode ser útil para várias coisas, desde anunciar o propósito do site até especificar alvos para traduções.

Os anúncios podem ser publicados em cada nível no menu *Manage*, usando `:guilabel:'Publicar anúncio'`:






 Weblate


Dashboard

Projects ▾

Languages ▾

Checks ▾



 WeblateOrg

translated 90%

Translations will be used only if they reach 60%. 

Components

Languages

Info

Search

Insights ▾

Files ▾

Tools ▾

Manage ▾

Share ▾

 Not watching ▾

Post announcement 

Message

You can use Markdown and mention users by @username.

Category

Info (light blue) ▾

Category defines color used for the message.

Expiry date

mm/dd/yyyy 

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ Notify users

The message is shown for all translations within the project, until its given expiry, or permanently until it is deleted.

Add

Powered by Weblate 4.16 [About Weblate](#) [Legal](#) [Contact](#) [Documentation](#) [Donate to Weblate](#)

Ele também pode ser adicionado usando a interface administrativa:

Weblate administration
WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

Home · Weblate translations · Announcements · Add Announcement

Add Announcement

Required fields are marked in bold.

Message:

Translations will be used only if they reach 60%.

You can use Markdown and mention users by @username.

Project: WeblateOrg

Component:

Language:

Category: Info (light blue)

Category defines color used for the message.

Expiry date: Today

The message will be not shown after this date. Use it to announce string freeze and translation deadline for next release.

☒ Notify users

Save and add another
Save and continue editing
SAVE

Os anúncios são então mostrados com base no seu contexto específico:

Nenhum contexto especificado

Mostrado no painel (página de chegada).

Projeto especificado

Mostrado dentro do projeto, incluindo todos os seus componentes e traduções.

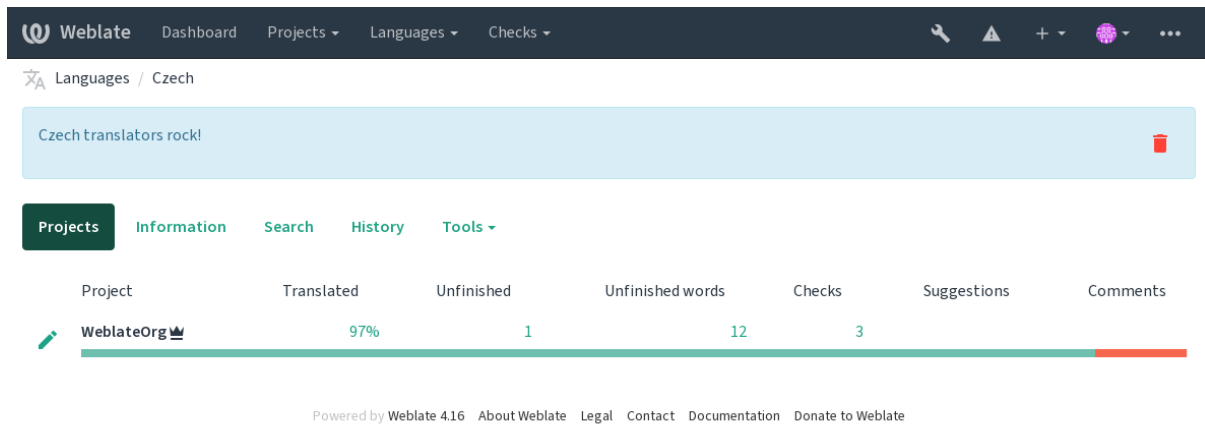
Componente especificado

Mostrado para um determinado componente e todas as suas traduções.

Idioma especificado

Mostrado na visão geral do idioma e todas as traduções nesse idioma.

Esta é a aparência na página de visão geral do idioma:



2.20 Lista de componentes

Especifique múltiplas listas de componentes para aparecer como opções no painel do usuário, a partir do qual os usuários podem selecionar uma visualização como exibição padrão. Veja [Painel](#) para saber mais.

Alterado na versão 2.20: Um status vai ser apresentado para cada componente listado no painel.

Os nomes e conteúdos das listas de componentes podem ser especificados na interface administrativa, na seção *Component lists*. Cada lista de componentes deve ter um nome que é exibido para o usuário e uma slug representando-a na URL.

Alterado na versão 2.13: Altera as configurações de painel para usuários anônimos a partir da interface administrativa, alterando qual painel é apresentado para usuários não autenticados.

2.20.1 Listas de componentes automáticas

Novo na versão 2.13.

Adicione componentes à lista automaticamente com base em suas slug criando regras *Automatic component list assignment*.

- Útil para manutenção de listas de componentes para instalações de grandes dimensões, ou no caso de você querer uma lista de componentes com todos os componentes em sua instalação de Weblate.

Dica: Faça uma lista de componentes contendo todos os componentes da sua instalação Weblate.

1. Defina *Automatic component list assignment* with `^.*$` as regular expression in both the project and the component fields, as shown on this image:

Weblate administration

WELCOME, **WEBLATE TEST**. [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

[Home](#) > [Weblate translations](#) > [Component lists](#) > [Add Component list](#)

Add Component list

Required fields are marked in bold.

Component list name:

All components

Display name

URL slug:

all-components

Name used in URLs and filenames.

☒ Show on dashboard

When enabled this component list will be shown as a tab on the dashboard

Components:

Available components ⓘ

Q

Filter

WeblateOrg/Django

WeblateOrg/Language names

WeblateOrg/WeblateOrg

Choose all ⓘ

Chosen components ⓘ

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

AUTOMATIC COMPONENT LIST ASSIGNMENTS

PROJECT REGULAR EXPRESSION ⓘ	COMPONENT REGULAR EXPRESSION ⓘ	DELETE? ⓘ
<div>^.*\$</div>	<div>^.*\$</div>	<div>✕</div>
<div><div>+</div> Add another Automatic component list assignment</div>		

Save and add another

Save and continue editing

SAVE

2.21 Módulos opcionais do Weblate

Vários módulos opcionais estão disponíveis para sua configuração.

402

Capítulo 2. Documentação de administrador

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` aos aplicativos instalados 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 habilitado) requer um token de API que pode ser obtido em seu *perfil de usuário*:

```
git clone 'https://user:KEY@example.org/git/weblate/main/'
```

Dica: Por padrão, os membros ou o grupo *Usuários* e usuário anônimo têm acesso aos repositórios para projetos públicos via *Acessar repositório* e funções de *Usuário avançado*.

2.21.2 Cobrança

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, você pode controlar a cobrança na interface de administração. Os usuários com cobrança habilitada obterão a nova aba *Cobrança* em seu *Perfil do usuário*.

O módulo de cobrança também permite que os administradores do projeto criem novos projetos e componentes sem serem superusuários (veja `add-projects`). Isso é possível quando as seguintes condições são atendidas:

- A cobrança está em seus limites configurados (qualquer uso excessivo resulta no bloqueio da criação do projeto/componente) e pago (se seu preço for diferente de zero)
- O usuário é administrador do projeto existente com cobrança ou o usuário é proprietário da cobrança (este último é necessário ao criar uma nova cobrança para que os usuários possam importar novos projetos).

Após a criação do projeto, o usuário 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 você 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

Ao alterar os documentos dos termos de serviço, ajuste `LEGAL_TOS_DATE` para que os usuários sejam forçados a concordar com os documentos atualizados.

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 você, 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 usuário do Weblate.

2.21.4 Avatares

Os avatares são baixados e armazenados em cache no lado do servidor para reduzir o vazamento de informações para os sites que os servem por padrão. 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

Ver também:

Cache de avatares, `AVATAR_URL_PREFIX`, `ENABLE_AVATARS`

2.21.5 Proteção contra spam

Você pode se proteger contra spam por usuários 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 usuário não autenticados
- Descrições e links de projetos e componentes

Nota: Este (entre outras coisas) depende do endereço IP do cliente, veja *Executando por trás de um proxy reverso* para configurá-lo apropriadamente.

Ver também:

Executando por trás de um proxy reverso, `AKISMET_API_KEY`, `WEBLATE_AKISMET_API_KEY`

2.21.6 Assinando commits do Git com GnuPG

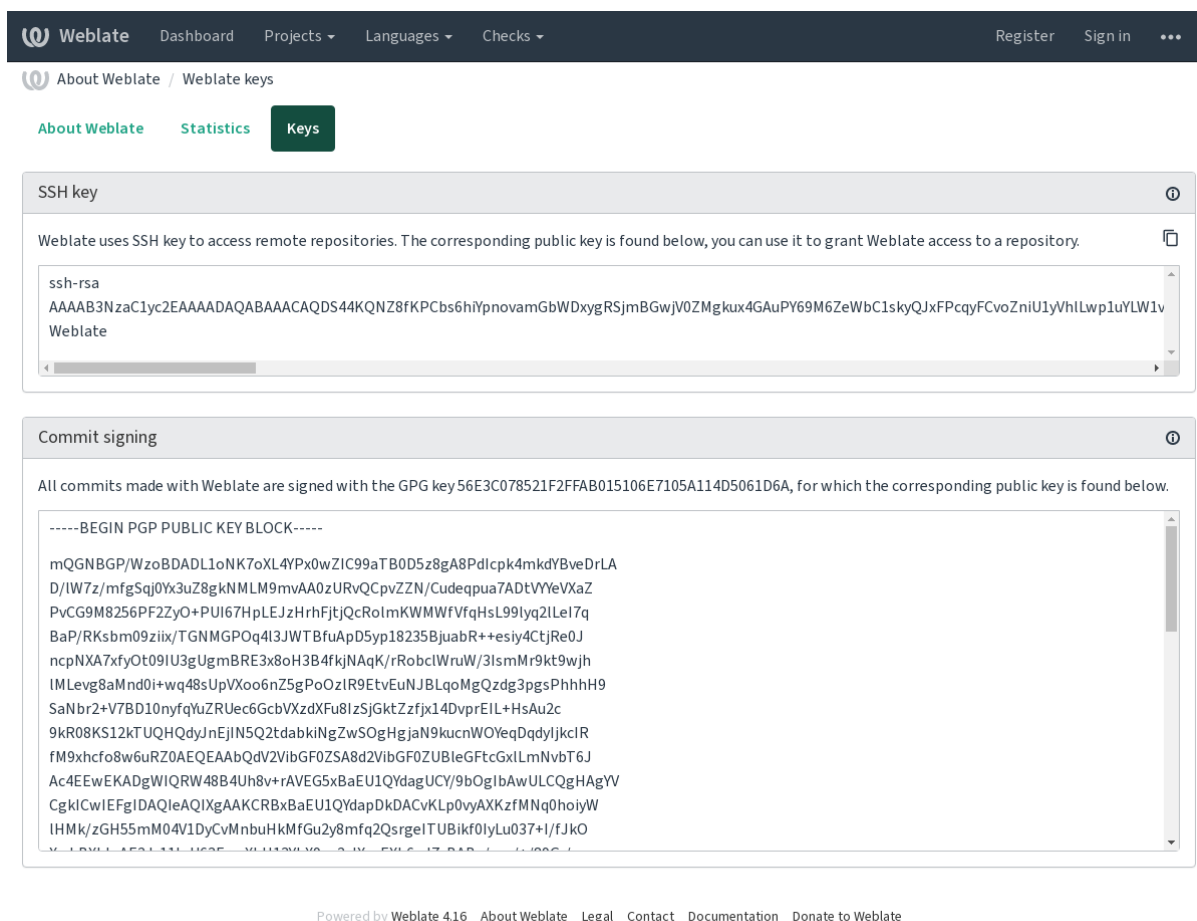
Novo na versão 3.1.

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.

Você pode encontrar a chave em `DATA_DIR` e a chave pública é mostrada na página “Sobre”:



The screenshot shows the Weblate web interface. At the top, there's a navigation bar with 'Weblate' logo, 'Dashboard', 'Projects', 'Languages', 'Checks', 'Register', 'Sign in', and a menu icon. Below the navigation bar, there's a breadcrumb trail 'About Weblate / Weblate keys'. The main content area has three tabs: 'About Weblate', 'Statistics', and 'Keys'. The 'Keys' tab is active. It contains two sections: 'SSH key' and 'Commit signing'. The 'SSH key' section has a title bar, a description, and a text area containing an SSH public key. The 'Commit signing' section has a title bar, a description, and a text area containing a GPG public key block.

SSH key

Weblate uses SSH key to access remote repositories. The corresponding public key is found below, you can use it to grant Weblate access to a repository.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDS44KQNZ8fKPCbs6hiYpnovamGbWdxygRSjmBGWjV0ZMgkux4GAuPY69M6ZeWbC1skyQJxFPcqyFCvoZniU1yVhLwpluYlW1v
Weblate
```

Commit signing

All commits made with Weblate are signed with the GPG key 56E3C078521F2FFAB015106E7105A114D5061D6A, for which the corresponding public key is found below.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGP/WzoBDADL1oNK7oXL4YPx0wZIC99aTB0D5z8gA8Pdlcpk4mkdYBveDrLA
D/IW7z/mfgSqj0Yx3uZ8gkNMLM9mvAA0zURvQCpvZZN/Cudeqqua7ADtVYeVXaZ
PvCG9M8256PF2ZyO+PUI67HpLEJzHrhFjtjQcRoImKWMWfVfqHsL99lyq2lLeI7q
BaP/RKsbm09ziix/TGNMGPOq4l3JWtBfuApD5yp18235BjuabR++esiy4CtjRe0J
ncpNXA7xfyOt09IU3gUgmBRE3x8oH3B4fkjNAqK/rRobclWruW/3lsmMr9kt9wjh
lMLevg8aMnd0i+wq48sUpVXoo6nZ5gPoOzIR9EtvEuNJBLoMgQzdg3pgsPhhhH9
SaNbr2+V7BD10nyfqYuZRUecGcbVXzdXFu8IzSjGktZzfjx14DvprEIL+HsAu2c
9kR08KS12kTUQHqdyJnEjIN5Q2tdabkiNgZwSOGHgjaN9kucnWOYeqDqdyIjklR
fM9xhcf08w6uRZ0AEQEAAbQdV2VibGF0ZSA8d2VibGF0ZUBleGFtcGxlLmNvbT6J
Ac4EEwEKADgWlQRW48B4U8v+rAVEG5xBaEU1QYdagUCY/9bOglbAwULCQgHAgYV
CgklCwIEFgIDAQIeAQIXgAAKCRBxBaEU1QYdapDkDACvKLp0vyAXKzfMNq0hoiyW
IHMk/zGH55mM04V1DyCvMnbuHkMfGu2y8mfq2QsrgeITUBikf0lyLu037+I/fJkO
-----END PGP PUBLIC KEY BLOCK-----
```

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

2. Alternatively you can also import existing keys into Weblate, just set `HOME=$DATA_DIR/home` when invoking `gpg`.

Ver também:

`WEBLATE_GPG_IDENTITY`

2.21.7 Limitação de taxa

Alterado na versão 3.2: A limitação de taxa agora aceita uma configuração mais refinada.

Alterado na versão 4.6: A limitação de taxa não se aplica mais a superusuários.

Várias operações no Weblate são limitadas por taxas. No máximo `RATELIMIT_ATTEMPTS` tentativas são permitidas dentro de `RATELIMIT_WINDOW` segundos. O usuário 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	Escopo	Tentativas permitidas	Janela de limite de tempo	Período de bloqueio
Registro	REGISTRATION	5	300	600
Enviando mensagem para administradores	MESSAGE	2	300	600
Autenticação de senha ao entrar	LOGIN	5	300	600
Pesquisa em todo o site	SEARCH	6	60	60
Traduzindo	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

Se um usuário falhar ao fazer o login `AUTH_LOCK_ATTEMPTS` vezes, a autenticação da senha será desativada na conta até ter passado pelo processo de redefinição da senha.

As configurações também podem ser aplicadas no contêiner 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](#).

Ver também:

[Limitação de taxa](#), [Executando 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. Você 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 Personalizando o Weblate

Amplie e personalize usando Django e Python. Contribua suas alterações para o upstream acima para que todos possam se beneficiar. Isso reduz seus custos de manutenção; código no Weblate é cuidado ao alterar interfaces internas ou refatorar o código.

Aviso: Nem interfaces internas nem modelos são considerados uma API estável. Por favor, revise suas próprias personalizações para cada atualização, as interfaces ou sua semântica podem mudar sem aviso prévio.

Ver também:

[Contribuindo para o Weblate](#)

2.22.1 Criando um módulo Python

Se você não está familiarizado com Python, você pode querer olhar para [Python For Beginners](#), explicando o básico e apontando para os tutoriais adicionais.

Para escrever um arquivo com 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.

Novo na versão 3.8-5: Quando se está *usando Docker* você pode colocar os módulos Python em `/app/data/python/` (veja *Volumes de contêiner Docker*), de forma que eles podem ser carregados pelo Weblate. Por exemplo, a partir de um *arquivo de substituição de configurações*.

Melhor ainda, transforme sua personalização em um pacote Python adequado:

1. Crie uma pasta para o seu pacote (usaremos `weblate_customization`).
2. Dentro dele, crie um arquivo `setup.py` para descrever o pacote:

```
from setuptools import setup

setup(
    name="weblate_customization",
    version="0.0.1",
    author="Your name",
    author_email="yourname@example.com",
    description="Sample Custom check for Weblate.",
    license="GPLv3+",
    keywords="Weblate check example",
    packages=["weblate_customization"],
)
```

3. Crie uma pasta para o módulo Python (também chamado de `weblate_customization`) para o código de personalização.
4. Dentro dele, crie um arquivo `__init__.py` para garantir que o Python possa importar o módulo.
5. Este pacote agora pode ser instalado usando `pip install -e`. Mais informações a serem encontradas em [Editable installs](#).
6. Uma vez instalado, o módulo pode ser usado na configuração Weblate (por exemplo, `weblate_customization.checks.FooCheck`).

Sua estrutura de pacotes deve ser assim:

```
weblate_customization
├── setup.py
└── weblate_customization
    ├── __init__.py
    ├── addons.py
    └── checks.py
```

Você pode encontrar um exemplo de personalização do Weblate em <https://github.com/WeblateOrg/customize-example>, ele abrange todos os tópicos descritos abaixo.

2.22.2 Alterando o logotipo

1. Crie um aplicativo Django simples contendo os arquivos estáticos que deseja substituir (veja *Criando um módulo Python*).

A marca aparece nos seguintes arquivos:

icons/weblate.svg

Logotipo mostrado na barra de navegação.

logo-*.png

Ícones web dependendo da resolução da tela e do navegador web.

favicon.ico

Ícone web usado por navegadores legados.

weblate-*.png

Avatares para bots ou usuários anônimos. Alguns navegadores web usam-nos como ícones de atalho.

email-logo.png

Usado em e-mails de notificações.

2. Adicione-o a `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # Add your customization as first
    "weblate_customization",
    # Weblate apps are here...
)
```

3. Execute `weblate collectstatic --noinput`, para coletar arquivos estáticos servidos aos clientes.

Ver também:

How to manage static files (e.g. images, JavaScript, CSS), *Servindo arquivos estáticos*

2.22.3 Verificações de qualidade personalizadas, extensões e correções automáticas

Para instalar seu código para *Correções automáticas personalizadas*, *Escrevendo as próprias verificações* ou *Escrevendo extensões* no Weblate:

1. Coloque os arquivos em seu módulo Python contendo a personalização ao Weblate (veja *Criando um módulo Python*).
2. Adicione seu caminho totalmente qualificado à classe Python nas configurações dedicadas (`WEBLATE_ADDONS`, `CHECK_LIST` ou `AUTOFIX_LIST`):

```
# Checks
CHECK_LIST += ("weblate_customization.checks.FooCheck",)

# Autofixes
AUTOFIX_LIST += ("weblate_customization.autofix.FooFixer",)

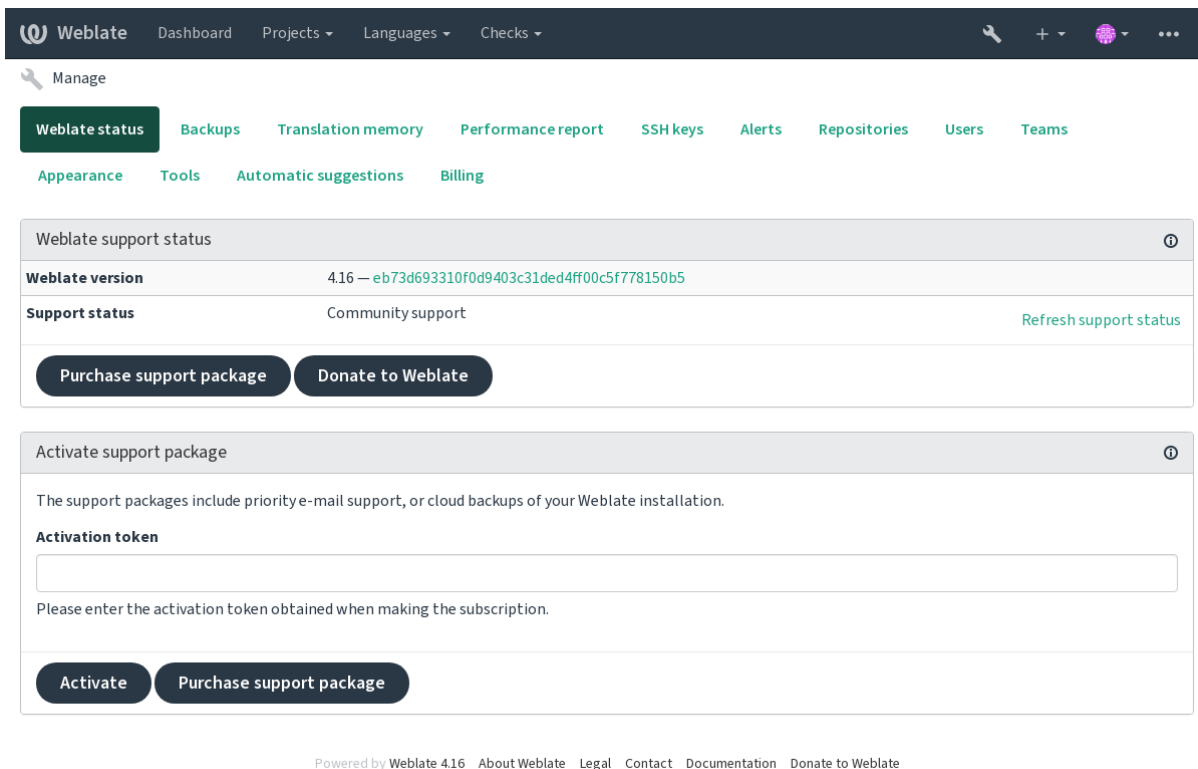
# Add-ons
WEBLATE_ADDONS += ("weblate_customization.addons.ExamplePreAddon",)
```

Ver também:

autocorreção-personalizada, verificações-próprias, extensão-própria, script-extensão

2.23 Interface de gerenciamento

A interface de gerenciamento oferece configurações de administração sob a URL `/manage/`. Está disponível para usuários que se inscrevem com privilégios administrativos, acessíveis usando o ícone da chave inglesa no canto superior direito:



Ela inclui uma visão geral básica do seu Weblate:

- Status de suporte, veja *Obtendo suporte para o Weblate*
- Backups, veja *Fazendo backup e movendo 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
- Gerenciamento de chaves SSH, veja *Repositórios SSH*
- Visão geral de alertas para todos os componentes, veja *alerts*

2.23.1 A interface administrativa do Django

Aviso: Use with caution as this is a low level interface. You should not need it in most cases as most things are comfortably approachable through Weblate UI or API.

Aqui você pode gerenciar objetos armazenados no banco de dados, tais como usuários, 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

User profiles

Verified e-mails

+ Add

Change

+ Add

Change

+ Add

Change

AUTH TOKEN

Tokens

+ Add

Change

AUTHENTICATION

Groups

Roles

Users

+ Add

Change

+ Add

Change

+ Add

Change

BILLING

Billing plans

Customer billings

Invoices

+ Add

Change

+ Add

Change

+ Add

Change

FONTS

Font groups

Fonts

+ Add

Change

+ Add

Change

LEGAL

TOS agreements

+ Add

Change

PERIODIC TASKS

Clocked

Crontabs

Intervals

Periodic tasks

Solar events

+ Add

Change

+ Add

Change

+ Add

Change

+ Add

Change

+ Add

Change

PYTHON SOCIAL AUTH

Associations

Nonces

User social auths

+ Add

Change

+ Add

Change

+ Add

Change

SCREENSHOTS

Screenshots

+ Add

Change

TRANSLATION MEMORY

Translation memory entries

+ Add

Change

WEBLATE CONFIGURATION

Settings

+ Add

Change

WEBLATE LANGUAGES

Languages

+ Add

Change

WEBLATE TRANSLATIONS

Announcements

Component lists

Components

Contributor agreements

Projects

+ Add

Change

+ Add

Change

+ Add

Change

+ Add

Change

+ Add

Change

Recent actions

My actions

None available

Na seção *Relatórios*, você pode verificar o status de seu site, ajustá-lo para produção ou gerenciar chaves SSH usadas para acessar *Acessando repositórios*.

Gerencie objetos de banco de dados sob qualquer uma das seções. A mais interessante é provavelmente *Traduções do Weblate*, onde você pode gerenciar projetos traduzíveis, veja *Configuração de projeto* e *Configuração de componente*.

Idiomas do Weblate detém as definições de idiomas, explicado melhor em *Definições de idioma*.

Adicionando um projeto

A adição de um projeto serve como contêiner para todos os componentes. Normalmente você cria um projeto para um software, ou livro (Veja *Configuração de projeto* para informações sobre parâmetros individuais):

Weblate administration

WELCOME, **WEBLATE TEST** · [RETURN TO WEBLATE](#) / [DOCUMENTATION](#) / [CHANGE PASSWORD](#) / [SIGN OUT](#)

[Home](#) · [Weblate translations](#) · [Projects](#) · [Add Project](#)

Add Project

Required fields are marked in bold.

Project name:

Display name

URL slug:

Name used in URLs and filenames.

Project website:

Main website of translated project.

Translation instructions:

You can use Markdown and mention users by @username.

☒ Set "Language-Team" header

Lets Weblate update the "Language-Team" file header of your project.

☒ Use shared translation memory

Uses the pool of shared translations between projects.

☒ Contribute to shared translation memory

Contributes to the pool of shared translations between projects.

Access control:

Protected

How to restrict access to this project is detailed in the documentation.

☐ Enable reviews

Requires dedicated reviewers to approve translations.

☐ Enable source reviews

Requires dedicated reviewers to approve source strings.

☒ Enable hooks

Whether to allow updating this repository by remote hooks.

Language aliases:

Comma-separated list of language code mappings, for example: en_GB:en,en_US:en

Machinery settings:

Save and add another

Save and continue editing

SAVE

Ver também:

Configuração de projeto

Componentes bilíngues

Uma vez adicionado um projeto, os componentes de tradução podem ser adicionados a ele. (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

[illegible]

Ver também:

Configuração de componente, Formatos bilíngues e monolíngues

Componentes monolíngues

Para facilitar a tradução destes, forneça um arquivo de modelo contendo o mapeamento de IDs de mensagem para seu respectivo idioma fonte (geralmente inglês). (Ver *Configuração de componente* para obter informações sobre parâmetros individuais):

Ver também:*Configuração de componente, Formatos bilíngues e monolíngues*

2.24 Obtendo suporte para o Weblate

Weblate é um software livre protegido por copyleft e com suporte comunitário. Os assinantes recebem suporte prioritário sem custo adicional. Pacotes de ajuda pré-pago estão disponíveis para todos. Você pode encontrar mais informações sobre as ofertas de suporte atuais em <<https://weblate.org/support/>>.

2.24.1 Integrando suporte

Novo na versão 3.8.

Os pacotes de suporte adquiridos podem ser integrados opcionalmente ao seu [gerenciamento de assinatura](#) do Weblate, de onde você encontrará um link para ele. Detalhes básicos da instância sobre sua instalação também são relatados de volta ao Weblate desta forma.

The screenshot shows the Weblate dashboard interface. At the top, there's a navigation bar with 'Weblate' logo and links to 'Dashboard', 'Projects', 'Languages', and 'Checks'. Below this is a 'Manage' section with various tabs: 'Weblate status' (active), 'Backups', 'Translation memory', 'Performance report', 'SSH keys', 'Alerts', 'Repositories', 'Users', and 'Teams'. Under 'Weblate status', there are sub-tabs: 'Appearance', 'Tools', 'Automatic suggestions', and 'Billing'.

The main content area is divided into two sections:

- Weblate support status:** This section shows the current 'Weblate version' as 4.16 with a unique hash. The 'Support status' is 'Community support'. There are buttons for 'Purchase support package' and 'Donate to Weblate'. A 'Refresh support status' link is also present.
- Activate support package:** This section explains that support packages include priority e-mail support or cloud backups. It features an 'Activation token' input field and a note to enter the token obtained during subscription. Buttons for 'Activate' and 'Purchase support package' are at the bottom.

At the very bottom, a footer line reads: 'Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate'.

2.24.2 Dados enviados para o Weblate

- URL onde sua instância do Weblate está configurada
- Título do seu site
- A versão do Weblate que você está executando
- Contagem de alguns objetos em seu banco de dados Weblate (projetos, componentes, idiomas, textos fonte e usuários)
- A chave pública SSH da sua instância

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 suporte ainda é válido
- *Armazenamento de backup provisionado do Weblate*
- *Descubra Weblate*

Dica: Os pacotes de suporte adquiridos já estão ativados no momento da compra e podem ser usados sem integrá-los.

2.24.4 Descubra Weblate

Novo na versão 4.5.2.

Nota: Este recurso está atualmente em um beta inicial.

Descubra Weblate é um serviço opcional que facilita encontrar comunidades e servidores Weblate. Os usuários 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 para o Weblate*.

Para listar o seu servidor com uma assinatura de suporte ativa (veja *Integrando suporte*) no Descubra Weblate tudo que você precisa fazer é ativar isso no painel de gerenciamento:

Weblate
Dashboard
Projects
Languages
Checks

Manage

Weblate status
Backups
Translation memory
Performance report
SSH keys
Alerts
Repositories
Users
Teams

Appearance
Tools
Automatic suggestions
Billing

Weblate support status

Weblate version
4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5

Support status
Community support
Refresh support status

Discover Weblate
Your Weblate is not listed on weblate.org
Browse discovery

Enable discovery

Manage support package

Purchase support package

Donate to Weblate

Activate support package

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate

Purchase support package

Powered by Weblate 4.16
About Weblate
Legal
Contact
Documentation
Donate to Weblate

Listando seu servidor sem uma assinatura de suporte no Descubra Weblate:

1. Registre-se em <<https://weblate.org/user/>>
2. Registre seu servidor Weblate no banco de dados de descoberta em <<https://weblate.org/subscription/discovery/>>
3. Confirme a ativação do serviço em seu Weblate e ative a lista de descoberta em sua página de gerenciamento do Weblate usando o botão *Habilitar descoberta*:

Manage

Weblate status Backups Translation memory Performance report SSH keys Alerts Repositories Users Teams

Appearance Tools Automatic suggestions Billing

Weblate support status ⓘ

Weblate version 4.16 — eb73d693310f0d9403c31ded4ff00c5f778150b5

Support status Community support Refresh support status

Discover Weblate Your Weblate is not listed on weblate.org Browse discovery

Enable discovery

Manage support package Purchase support package Donate to Weblate

Activate support package ⓘ

The support packages include priority e-mail support, or cloud backups of your Weblate installation.

Activation token

Please enter the activation token obtained when making the subscription.

Activate Purchase support package

Powered by Weblate 4.16 About Weblate Legal Contact Documentation Donate to Weblate

Personalizando a listagem

Você 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 você encontrará várias informações legais que você pode precisar para operar Weblate em certas jurisdições legais. É fornecido como um meio de orientação, sem qualquer garantia de precisão ou correção. Em última análise, é sua responsabilidade garantir que seu uso do Weblate esteja em conformidade com todas as leis e regulamentos aplicáveis.

2.25.1 Conformidade de licenciamento

O Weblate vem com especificação de licença compatível com [REUSE 3.0](#).

2.25.2 ITAR e outros controles de exportação

O Weblate pode ser usado dentro de seu próprio datacenter ou nuvem privada virtual. Como tal, ele pode ser usado para armazenar informações ITAR ou outras controladas por exportação; no entanto, os usuários finais são responsáveis por garantir tal conformidade.

Hosted Weblate serviço não foi auditado pela conformidade com ITAR ou outros controles de exportação, e atualmente não oferece a capacidade de restringir traduções de acesso por país.

2.25.3 Controles de criptografia dos EUA

O Weblate não contém nenhum código criptográfico, mas pode ser objeto de controles de exportação, pois usa componentes de terceiros utilizando criptografia para autenticação, integridade de dados e confidencialidade.

Provavelmente Weblate seria classificado como ECCN 5D002 ou 5D992 e, como software livre publicamente disponível, não deve ser sujeito ao EAR (veja “Itens de criptografia NÃO estão sujeitos a EAR <<https://www.bis.doc.gov/index.php/policy-guidance/encryption/1-encryption-items-not-subject-to-the-ear>>”).

Componentes de software utilizados por Weblate (listando somente os componentes relacionados à função criptográfica):

Python

Veja https://wiki.python.org/moin/PythonSoftwareFoundationLicenseFaq#Is_Python_subject_to_export_laws.3F

GnuPG

Opcionalmente usado pelo Weblate

Git

Opcionalmente usado pelo Weblate

curl

Usado pelo Git

OpenSSL

Usado pelo Python e cURL

A força de chaves de criptografia depende da configuração do Weblate e os componentes de terceiros que interage com ele, mas em qualquer decente instalação, ele irá incluir todas funções criptográficas com exportação restrita:

- Em excesso de 56 bits para um algoritmo simétrico
- Fatorização de inteiros acima de 512 bits para um algoritmo assimétrico
- Cálculo de logaritmos discretos em um grupo multiplicativo de um campo finito de tamanho maior do que 512 bits para um algoritmo assimétrico
- Logaritmos discretos em um grupo diferente do que acima de 112 bits para um algoritmo assimétrico

O Weblate não tem nenhum recurso de ativação criptográfica, mas pode ser configurado de uma maneira onde nenhum código de criptografia estaria envolvido. Os recursos criptográficos incluem:

- Acessar servidores remotos usando protocolos seguros (HTTPS)
- Gerar assinaturas para commits de código (PGP)

Ver também:

[Controles de Exportação \(EAR\) em Software de Código Aberto](#) (*inglês*)

3.1 Contribuindo para o Weblate

Há dezenas de maneiras de melhorar o Weblate. Você pode escolher uma com a qual você se sente confortável, seja codificação, design gráfico, documentação, patrocínio ou uma ideia:

- *Relatando problemas no Weblate*
- *Começando a contribuir código para o Weblate*
- *Contribuindo para módulos do Weblate*
- *Traduzindo o Weblate*
- *Contribua para a documentação do Weblate*
- *Discussões do Weblate*
- *Financiando o desenvolvimento do Weblate*

3.1.1 Traduzindo o Weblate

O Weblate está sendo continuamente [traduzido](#) usando o próprio Weblate. Sinta-se à vontade para participar do esforço de disponibilizar o Weblate no maior número possível de idiomas humanos. Isso traz o Weblate para mais perto de seus usuários!

Se você encontrar um possível erro no texto fonte, pode marcá-lo com um comentário no editor Weblate. Dessa forma, pode ser discutido e corrigido. Se você tiver certeza, também pode clicar no link na seção *Local do texto fonte* e enviar um PR com sua correção.

3.1.2 Contribua para a documentação do Weblate

Você é bem-vindo para melhorar a página de documentação de sua escolha. Faça isso facilmente clicando no botão *Editar no GitHub* no canto superior direito da página.

Respeite essas diretrizes ao escrever:

1. Não remova parte da documentação se ela for válida.
2. Use uma linguagem clara e de fácil compreensão. Você está escrevendo documentos técnicos, não um poema. Nem todos os leitores de documentos são falantes nativos, fique atento.
3. Não tenha medo de perguntar se você não tem certeza. Se você tiver que perguntar sobre algum recurso durante a edição, não altere seus documentos antes de ter a resposta. Isso significa: você muda ou pergunta. Não faça os dois ao mesmo tempo.
4. Verifique suas alterações executando as ações descritas ao seguir os documentos.
5. Envie PR com alterações em pequenos pedaços para tornar mais fácil e rápido revisar e mesclar.
6. Se você quiser reescrever e alterar a estrutura de um grande artigo, faça isso em duas etapas:
 1. Reescreva
 2. Depois que a reescrita for revisada, polida e mesclada, altere a estrutura dos parágrafos em outro PR.

Dica: Você pode [traduzir os documentos](#).

3.1.3 Estendendo definições embutidas de idioma

As definições de idioma estão no [repositório language-data](#) do Weblate.

Você está convidado a adicionar definições de idioma em falta a `languages.csv`, outros arquivos são gerados a partir desse arquivo.

3.1.4 Discussões do Weblate

Se você tiver uma ideia e não tiver certeza se ela é adequada para um problema, não se preocupe. Você pode entrar na comunidade de [discussões do GitHub](#).

3.1.5 Financiando o desenvolvimento do Weblate

Você pode impulsionar o desenvolvimento do Weblate na [página de doação](#). Os fundos coletados lá são usados para permitir hospedagem de 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 você obtém como um financiador orgulhoso.

Apoiadores que financiaram o Weblate

Lista de apoiadores do Weblate:

- Yashiro Ccs
- Cheng-Chia Tseng
- Timon Reinhard
- [Cassidy James](#)
- Loic Dachary

- [Marozed](#)
- <https://freedombox.org/>
- [GNU Solidario \(GNU Health\)](#)
- [BallotReady](#)
- [Richard Nespithal](#)
- [MyExpenses.Mobi](#)
- [Michael Totschnig](#)

Gostaria de estar na lista? Veja as opções no [Doar para o Weblate](#).

3.2 Começando a contribuir código para o Weblate

Entenda o código-fonte do Weblate passando por *Código-fonte do Weblate*, *Frontend do Weblate* e *Componentes internos do Weblate*.

3.2.1 Começando com a base de código

Familiarize-se com a base de código do Weblate, dando uma olhada nos bugs etiquetados como [good first issue](#).

Você é muito bem-vindo(a) a começar a trabalhar nessas questões sem perguntar. Apenas anuncie isso no relatório de problema, para que fique claro que alguém está trabalhando nesse relatório de problema.

3.2.2 Executando Weblate localmente

A abordagem mais confortável para começar a desenvolver Weblate é seguir `.../admin/install/source`. Isso lhe dará um `virtualenv` com fontes editáveis do Weblate.

1. Clone o código-fonte do Weblate:

```
git clone https://github.com/WeblateOrg/weblate.git
cd weblate
```

2. Crie um `virtualenv`:

```
virtualenv .venv
.venv/bin/activate
```

3. Instale Weblate (para isso, você vai precisar de algumas dependências do sistema, veja `.../admin/install/source`):

```
pip install -e .
```

3. Instale todas as dependências úteis para o desenvolvimento:

```
pip install -r requirements-dev.txt
```

4. Inicie um servidor de desenvolvimento:

```
weblate runserver
```

5. Dependendo da sua configuração, você também pode querer iniciar workers do Celery:

```
./weblate/examples/celery start
```

6. Para executar um teste (veja [Teste local](#) para mais detalhes):

```
. scripts/test-database.sh
./manage.py test
```

Ver também:

Instalando a partir do código-fonte

3.2.3 Executando Weblate localmente no Docker

Se você tiver Docker e docker-compose instalados, você pode colocar para funcionar o ambiente de desenvolvimento simplesmente executando:

```
./rundev.sh
```

Isso irá criar uma imagem do Docker de desenvolvimento e iniciá-lo. Weblate está sendo executado em `<http://127.0.0.1:8080/>` e você pode entrar como o usuário `admin`, usando `admin` como a senha. A nova instalação está vazia, então você pode querer seguir *Adicionando projetos e componentes de tradução*.

Os arquivos `Dockerfile` e `docker-compose.yml` para isso estão localizados no diretório `dev-docker`.

O script também aceita alguns parâmetros; para executar testes, execute-o com o parâmetro `test` e então especifique qualquer parâmetro `test`, por exemplo executar apenas testes no módulo `weblate.machine`:

```
./rundev.sh test --failfast weblate.machine
```

Nota: Tenha cuidado para que os seus contêineres Docker estejam prontos e em execução antes de fazer os testes. Você pode verificar isso executando o comando `docker ps`.

Para exibir os logs:

```
./rundev.sh logs
```

Para parar os contêineres em segundo fundo, execute:

```
./rundev.sh stop
```

A execução do script sem argumentos vai recriar o contêiner Docker e reiniciá-lo.

Nota: Esta não é uma configuração adequada para a produção, pois inclui vários hacks que são inseguros, mas que tornam o desenvolvimento mais fácil.

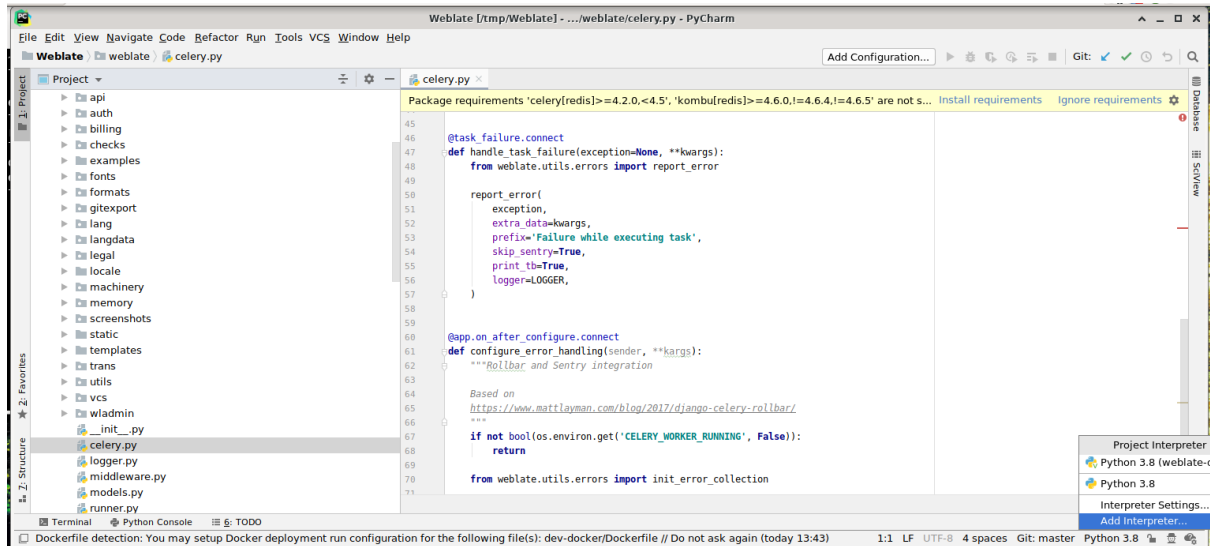
3.2.4 Inicializando sua instância de desenvolvimento

Você pode querer usar `import_demo` para criar traduções de demonstração e `createadmin` para criar um usuário administrador.

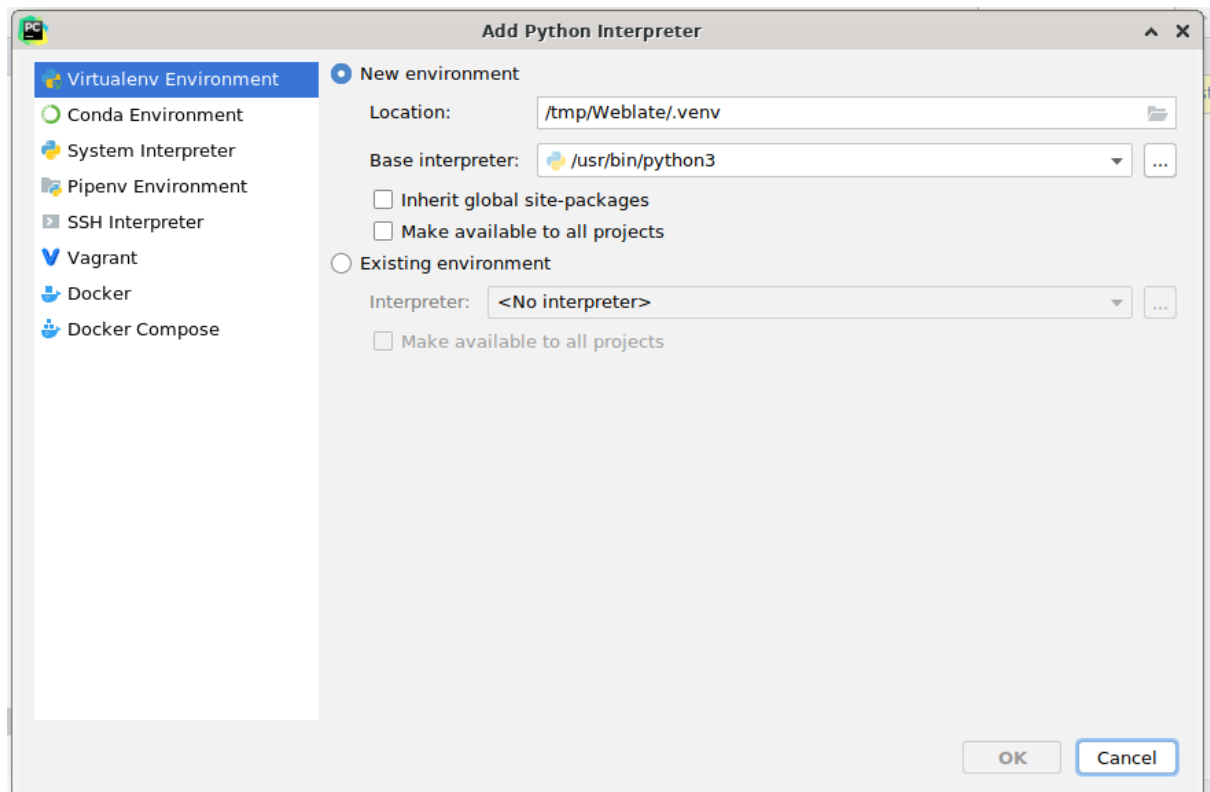
3.2.5 Codificando Weblate com PyCharm

PyCharm é uma IDE conhecida para Python, aqui estão algumas diretrizes para lhe ajudar a configurar seu projeto Weblate nele.

Considerando que você acabou de clonar o repositório GitHub, para uma pasta, basta abri-la com o PyCharm. Assim que a IDE estiver aberta, o primeiro passo é especificar o interpretador que você deseja:



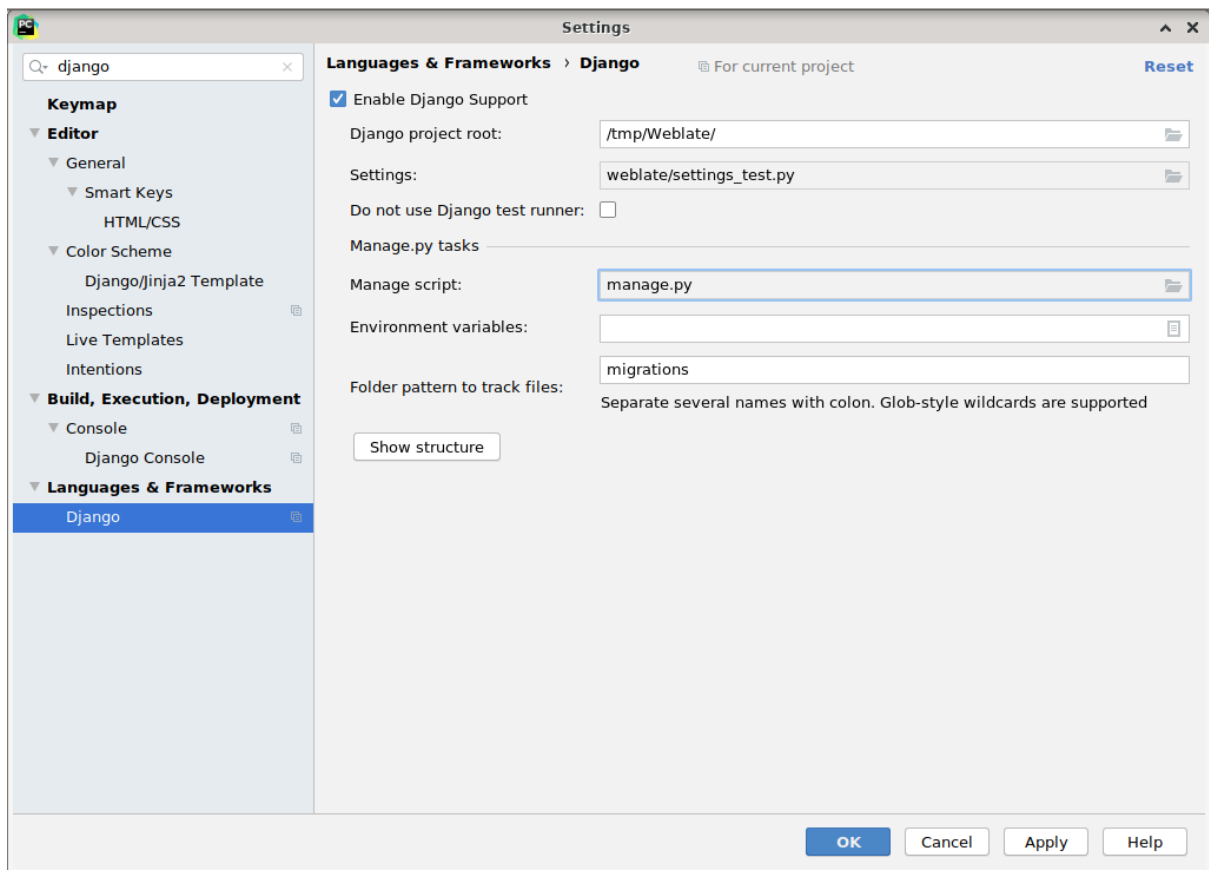
Você pode optar por deixar PyCharm criar o virtualenv para você, ou selecionar um já existente:



Não se esqueça de instalar as dependências quando o interpretador estiver configurado: faça isso através do console (o console do IDE vai usar diretamente o seu virtualenv por padrão, ou através da interface quando receber um aviso sobre dependências em falta).

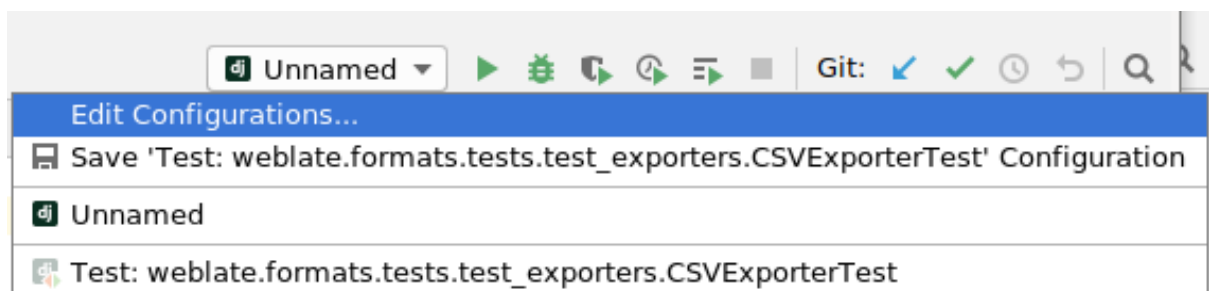
O segundo passo é definir a informação certa para usar o Django nativo dentro do PyCharm: a ideia é ser capaz de acionar imediatamente os testes unitários no IDE. Para isso, você precisa especificar o caminho raiz do projeto

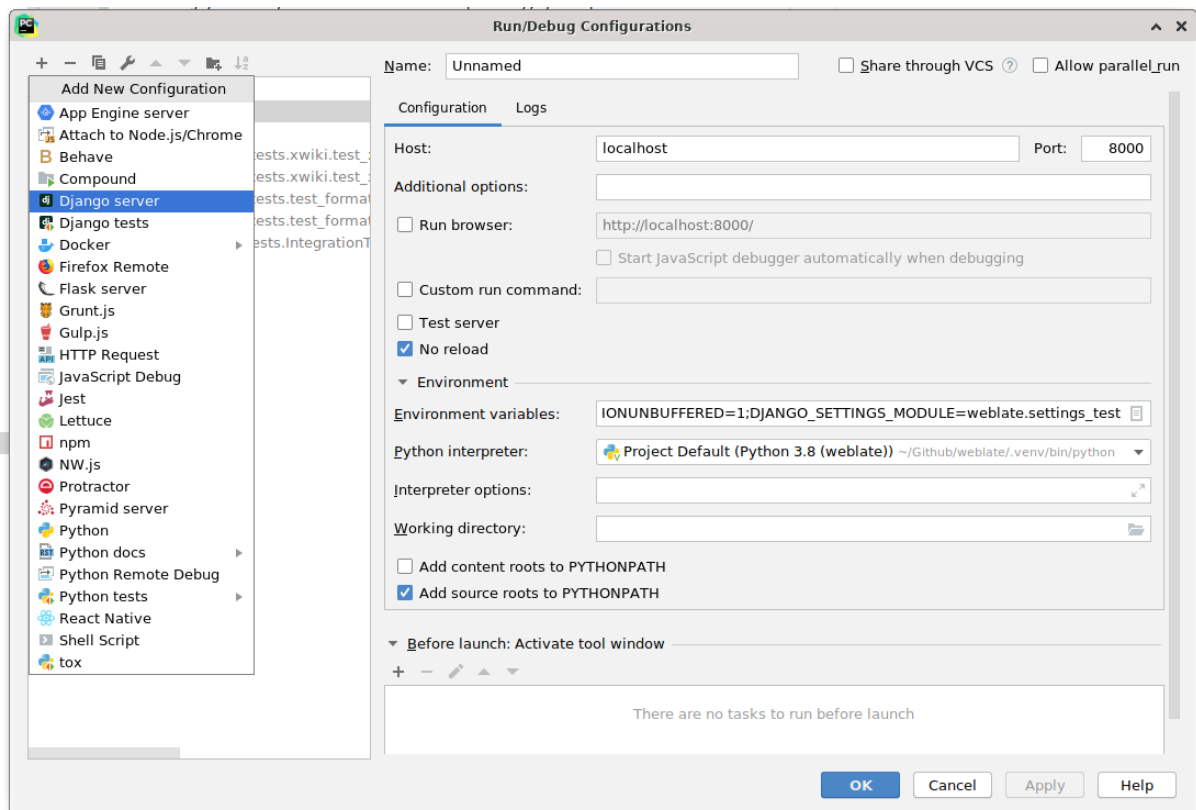
Django e o caminho para suas configurações:



Tenha cuidado, *Django project root* é a raiz real do repositório, não o subdiretório do Weblate. Sobre as configurações, você poderia usar o `weblate/settings_test.py` do repositório, mas você pode criar sua própria configuração e configurá-la lá.

O último passo é executar o servidor e colocar pontos de interrupção no código para poder depurá-lo. Isto é feito criando uma nova configuração do *Django Server*:





Dica: Tenha cuidado com a propriedade chamada *No reload*: ela evita do servidor ser recarregado automaticamente quando você modificar os arquivos. Isso permite que os pontos de interrupção de depuração existentes persistam, quando eles normalmente seriam descartados no recarregamento do servidor.

3.3 Código-fonte do Weblate

O Weblate é desenvolvido no [GitHub](#). Você é bem-vindo para criar um fork do código e abrir pull requests. Patches em qualquer outra forma são bem-vindos também.

Ver também:

Confira *Componentes internos do Weblate* para ver como o Weblate se parece por dentro.

3.3.1 Diretrizes de codificação

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 usuário.

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 usando o formatador de código **black**.

Para verificar a qualidade do código, você pode usar **flake8**, os plugins recomendados estão listados em `.pre-commit-config.yaml` e sua configuração é colocada em `setup.cfg`.

A abordagem mais fácil para impor tudo isso é instalar **pre-commit**. O repositório contém a configuração para verificar se os arquivos do commit estão sãos. Depois de instalá-lo (ele já está incluído no `requirements-lint.txt`), ative-o executando `pre-commit install` na sua cópia do Weblate. Desta forma, todas as suas alterações serão verificadas automaticamente.

Você também pode acionar a verificação manualmente, para verificar todos os arquivos executados:

```
pre-commit run --all
```

3.4 Depuração do Weblate

Os bugs podem se comportar como falhas no aplicativo ou como vários mau comportamentos. Você é bem-vindo a coletar informações sobre qualquer problema desse tipo e enviá-las para o [rastreador de problemas](#).

3.4.1 Modo de depuração

Ao 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 vaziar dados privados.

Em um ambiente de produção, use [ADMINS](#) para receber e-mails contendo relatórios de erros ou configurar a coleta de erros usando um serviço de terceiros.

Ver também:

[Desabilitar o modo de depuração](#), [Configurar corretamente administradores](#), [Coletando relatórios de erros](#)

3.4.2 Logs do Weblate

Weblate pode produzir registros (logs) detalhados do que está acontecendo em segundo fundo. Na configuração padrão, ele usa syslog e isso faz o log aparecer em `/var/log/messages` ou `/var/log/syslog` (dependendo da configuração do seu daemon syslog).

O processo de Celery (veja [Tarefas de fundo usando Celery](#)) normalmente também produz seus próprios registros. O exemplo de configurações de todo o sistema registra em vários arquivos em `/var/log/celery/`.

Os contêineres Docker registram a sua saída (como é habitual no mundo Docker), por isso pode olhar para os logs utilizando `docker-compose logs`.

Ver também:

[Amostra de configuração](#) contém uma configuração de `LOGGING`.

3.4.3 Não processando tarefas de segundo plano

Muitas coisas 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 usando Celery*
- Verifique o status da fila do Celery em *Interface de gerenciamento* ou usando `celery_queues`
- Procure erros nos registros do Celery (veja *Logs do Weblate*)

3.4.4 Não recebendo e-mails do Weblate

Você pode verificar se o e-mail de saída está funcionando corretamente usando o comando de gerenciamento `send-testemail` (veja *Invocando comandos de gerenciamento* para instruções sobre como invocá-lo em diferentes ambientes) ou usando *Interface de gerenciamento* sob a aba *Ferramentas*.

Estes enviam e-mails diretamente, para que isto verifique se a sua configuração SMTP está correta (veja *Configuração de e-mail de saída*). A maioria dos e-mails da Weblate são, no entanto, enviados em segundo plano e pode haver alguns problemas com o Celery envolvido também, veja *Não processando tarefas de segundo plano* para depuração disso.

3.4.5 Analisando falhas de aplicativo

Caso o aplicativo falhe, é útil recolher o máximo de informação possível sobre a falha. Isso pode ser feito usando serviços de terceiros que podem coletar tais informações automaticamente. Você pode encontrar informações sobre como configurar isso em *Coletando relatórios de erros*.

3.4.6 Falhas silenciosas

Muitas tarefas são descarregadas para o Celery para processamento em segundo plano. As falhas não são exibidas na interface do usuário, mas aparecem nos logs do Celery. A configuração em *Coletando relatórios de erros* ajuda a perceber essas falhas mais facilmente.

3.4.7 Problemas de desempenho

Caso o Weblate tenha um mau desempenho em algum cenário, por favor, recolha os registros relevantes mostrando o problema, e qualquer coisa que possa ajudar a descobrir onde o código pode ser melhorado.

Caso alguns pedidos demorem muito tempo sem nenhuma indicação, você pode querer instalar `dogslow` juntamente com *Coletando relatórios de erros* e obter tracebacks detalhados na ferramenta de coleta de erros.

Caso o desempenho lento esteja vinculado ao banco de dados, você também pode habilitar o log de todas as consultas do banco de dados usando a seguinte configuração após habilitar `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 de sua estrutura de código e é baseado no [Django](#).

3.5.1 Estrutura de diretórios

Visão geral rápida da estrutura de diretórios do repositório principal Weblate:

docs

Código-fonte desta documentação, que pode ser construída usando [Sphinx](#).

dev-docker

Código do Docker para executar o servidor de desenvolvimento, veja [Executando Weblate localmente no Docker](#).

weblate

Código-fonte do Weblate como um aplicativo [Django](#), veja [Componentes internos do Weblate](#).

weblate/static

Arquivos de cliente (CSS, Javascript e imagens), veja [Frontend do Weblate](#).

3.5.2 Módulos

O Weblate consiste em vários aplicativos Django (alguns opcionais, veja [Módulos opcionais do Weblate](#)):

accounts

Conta de usuário, perfis e notificações.

addons

Extensões para ajustar o comportamento do Weblate, veja [Extensões](#).

api

API baseada no [framework Django REST](#).

auth

Autenticação e permissões.

billing

O módulo opcional de [Cobrança](#).

checks

Módulo de [Verificações de qualidade](#) de texto de tradução.

fonts

Módulo de verificações de renderização de fonte.

formats

Camada de abstração de formato de arquivo baseada no [translate-toolkit](#).

gitexport

O módulo opcional [Exportador git](#).

lang

Módulo que define idioma e modelos plurais.

legal

O módulo opcional *Legal*.

machinery

Integração de serviços de tradução de máquina.

memory

Memória de tradução embutida, veja *Memória de tradução*.

screenshots

Gerenciamento de capturas de tela e módulo OCR.

trans

Módulo principal que trata das traduções.

utils

Vários utilitários auxiliares.

vcs

Abstração do sistema de controle de versão.

wladmin

Personalização da interface administrativa do Django.

3.6 Desenvolvendo extensões

Extensões são uma forma de personalizar o fluxo de trabalho de localização no Weblate.

class weblate.addons.base.**BaseAddon** (*storage=None*)

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 seu commit feito para o repositório.

post_push (*component*)

Gancho acionado após repositório ter o push feito para o upstream.

post_update (*component*, *previous_head*: *str*, *skip_push*: *bool*)

Gancho acionado após o repositório ser atualizado a partir do upstream.

Parâmetros

- **previous_head** (*str*) – HEAD do repositório antes da atualização, pode estar vazio na clonagem inicial.
- **skip_push** (*bool*) – Se a operação de extensão deve ignorar o push de alterações para o upstream. Normalmente você pode passar isso para métodos subjacentes como `commit_and_push` ou `commit_pending`.

pre_commit (*translation*, *author*)

Gancho acionado antes das alterações terem seu commit feito para o repositório.

pre_push (*component*)

Gancho acionado antes do repositório ter seu push feito para o upstream.

pre_update (*component*)

Hook acionado antes do repositório ser atualizado a partir do upstream.

save_state ()

Salva informações do estado da extensão.

store_post_load (*translation*, *store*)

Gancho acionado após um arquivo ser analisado.

Ele recebe uma instância de uma classe de formato de arquivo como um argumento.

Isso é útil para modificar os parâmetros de classe de formato de arquivo, por exemplo, ajustar como o arquivo será salvo.

unit_pre_create (*unit*)

Gancho acionado antes que a nova unidade seja criada.

Aqui está um exemplo de extensão:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

from django.utils.translation import gettext_lazy as _

from weblate.addons.base import BaseAddon
from weblate.addons.events import EVENT_PRE_COMMIT

class ExampleAddon(BaseAddon):
    # Filter for compatible components, every key is
    # matched against property of component
    compat = {"file_format": {"po", "po-mono"}}
    # List of events add-on should receive
    events = (EVENT_PRE_COMMIT,)
    # Add-on unique identifier
    name = "weblate.example.example"
    # Verbose name shown in the user interface
    verbose = _("Example add-on")
    # Detailed add-on description
    description = _("This add-on does nothing it is just an example.")

    # Callback to implement custom behavior
    def pre_commit(self, translation, author):
        return
```

3.7 Frontend do Weblate

O frontend atualmente é construído usando Bootstrap, jQuery e algumas bibliotecas de terceiros.

3.7.1 Navegadores suportados

Weblate oferece suporte às versões mais recentes e estáveis de todos os principais navegadores e plataformas.

Navegadores alternativos que usam a versão mais recente do WebKit, Blink ou Gecko, seja diretamente ou por meio da API de visualização da web da plataforma, não são explicitamente suportados. No entanto, o Weblate deve (na maioria dos casos) ser exibido e funcionar corretamente nesses navegadores também.

Navegadores mais antigos podem funcionar, mas alguns recursos podem ser limitados.

3.7.2 Gerenciamento de dependências

O gerenciador de pacotes yarn é usado para atualizar bibliotecas de terceiros. A configuração reside em `scripts/yarn` e há um script wrapper `scripts/yarn-update` para atualizar as bibliotecas, construí-las e copiá-las para os locais corretos em `weblate/static/vendor`, onde todo o terceiro código parcialmente frontend está localizado. O código específico do Weblate deve ser colocado diretamente em `weblate/static` ou em diretórios específicos 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 arquivos JavaScript e CSS.

Também usamos [ESLint](#) para verificar o código JavaScript.

3.7.4 Localização

Se você precisar de qualquer texto visível para o usuário no código do frontend, ele deve ser localizável. Na maioria dos casos, tudo que você precisa é envolver seu texto dentro da função `gettext`, mas existem recursos mais complexos disponíveis:

```
document.write(gettext('this is to be translated'));

var object_count = 1 // or 0, or 2, or 3, ...
s = 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'
```

Ver também:

[Tópico sobre tradução na documentação do Django](#)

3.7.5 Ícones

Weblate atualmente usa ícones de design de material. Caso você esteja procurando por um novo símbolo, verifique [Material Design Icons](#) ou [Material Design Resources](#).

Além disso, existe `scripts/optimize-svg` para reduzir o tamanho do SVG, já que a maioria dos ícones são embutidos no HTML para permitir estilização dos caminhos.

3.8 Relatando problemas no Weblate

O rastreador de problemas do Weblate está hospedado no GitHub.

Sinta-se à vontade para relatar quaisquer problemas que você tenha ou sugerir melhorias para o Weblate lá. Existem vários modelos preparados para lhe orientar confortavelmente durante o relatório de problemas.

Se o que você encontrou é um problema de segurança no Weblate, consulte a seção [Problemas de segurança](#) abaixo.

Se você não tem certeza sobre seu relatório de bug ou solicitação de recurso, você pode tentar [Discussões do Weblate](#).

3.8.1 Problemas de segurança

A fim de dar à comunidade tempo para responder e fazer upgrade, recomendamos enfaticamente que você relate todos os problemas de segurança em particular. O HackerOne é usado para lidar com problemas de segurança e pode ser relatado diretamente em [HackerOne](#). Depois de enviá-lo lá, a comunidade terá tempo limitado, mas suficiente para resolver o incidente.

Como alternativa, relate para security@weblate.org, que termina no HackerOne também.

Se você não quiser usar o HackerOne, por qualquer motivo, pode enviar o relatório por e-mail para mi-chal@weblate.org. Você pode optar por criptografá-lo usando esta chave PGP `3CB 1DF1 EF12 CF2A C0EE 5A32 9C27 B313 42B7 511D`. Você também pode obter a chave PGP em [Keybase](#).

Nota: O Weblate depende de componentes de terceiros para muitas coisas. Caso você encontre uma vulnerabilidade afetando um desses componentes em geral, relate-o diretamente ao respectivo projeto.

Alguns deles são:

- [Django](#)
 - [framework Django REST](#)
 - [Python Social Auth](#)
-

3.9 Conjunto de testes e integração contínua do Weblate

Existem conjuntos de teste para a maior parte do código atual, aumente a cobertura adicionando casos de teste para qualquer nova funcionalidade e verifique se funciona.

3.9.1 Integração contínua

Current test results can be found on [GitHub Actions](#) and coverage is reported on [Codecov](#).

Existem vários trabalhos para verificar diferentes aspectos:

- Testes de unidade
- Construção de documentação e links externos
- Testes de migração de todas as versões com suporte
- Linting de código
- Verificação de configuração (garante que os arquivos dist gerados não percam nada e possam ser testados)

A configuração do CI está no diretório `.github/workflows`. Ele usa muito scripts auxiliares armazenados no diretório `ci`. Os scripts também podem ser executados manualmente, mas eles requerem várias variáveis de ambiente, principalmente definindo o arquivo de configurações do Django para usar e conexão com o banco de dados. A definição de exemplo disso está em `scripts/test-database.sh`:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

# Simple way to configure test database from environment

# shellcheck shell=sh

# 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.sh
./ci/run-migrate
./ci/run-test
./ci/run-docs
```

3.9.2 Teste local

Para executar um conjunto de testes localmente, use:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test
```

Dica: Você precisará de um servidor de banco de dados (PostgreSQL) a ser usado para os testes. Por padrão, o Django cria um banco de dados separado para executar testes com o prefixo `test_`, então no caso de suas configurações estarem configuradas para usar `weblate`, os testes usarão o banco de dados `test_weblate`. Veja [Configuração de banco de dados para o Weblate](#) para instruções de configuração.

O `weblate/settings_test.py` é usado no ambiente CI também (veja [Integração contínua](#)) e pode ser ajustado usando variáveis de ambiente:

```
# Copyright © Michal Čihař <michal@weblate.org>
#
# SPDX-License-Identifier: GPL-3.0-or-later

# Simple way to configure test database from environment

# shellcheck shell=sh

# 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, você deve coletar arquivos estáticos, pois alguns testes contam com a presença deles:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py collectstatic
```

Você também pode especificar testes individuais para executar:

```
DJANGO_SETTINGS_MODULE=weblate.settings_test ./manage.py test weblate.gitexport
```

Dica: Os testes também podem ser executados dentro de um contêiner Docker do desenvolvedor, consulte [Executando Weblate localmente no Docker](#).

Ver também:

Veja [Testando em Django](#) para mais informações sobre a execução e escrita de testes para Django.

3.10 Esquemas de dados

Weblate usa [JSON Schema](#) para definir o layout de arquivos JSON externos.

3.10.1 Esquema de memória de tradução do Weblate

https://weblate.org/schemas/weblate-memory.schema.json	
tipo	array
itens	<i>O item da memória de tradução</i>
	tipo <i>objeto</i>
	propriedades
	<ul style="list-style-type: none"> category <i>A categoria do texto</i> 1 é global, 2 é compartilhado, 10000000+ são específicos de projeto, 20000000+ são específicos de usuário tipo <i>inteiro</i> exemplos 1 mínimo 0 padrão 1 origin <i>A origem do texto</i> Nome de arquivo ou nome de componente

continua na próxima página

Tabela 1 – continuação da página anterior

	tipo	<i>texto</i>
	exemplos	test.tmx projeto/componente
• source	padrão	
	<i>O texto fonte</i>	
	tipo	<i>texto</i>
	exemplos	Olá
• source_language	minLength	1
	padrão	
	<i>O idioma fonte</i>	
	ISO 639-1 / ISO 639-2 / IETF BCP 47	
• target	tipo	<i>texto</i>
	exemplos	en
	pattern	^[^]+\$
	padrão	
• target_language	<i>O texto de destino</i>	
	tipo	<i>texto</i>
	exemplos	Ahoj
	minLength	1
• target_language	padrão	
	<i>O idioma de destino</i>	
	ISO 639-1 / ISO 639-2 / IETF BCP 47	
	tipo	<i>texto</i>
	exemplos	cs
	pattern	^[^]+\$
	padrão	
	additionalProperties	False
definições		

Ver também:

Memória de tradução, *dump_memory*, *import_memory*

3.10.2 Exportação de dados de usuário do Weblate

https://weblate.org/schemas/weblate-userdata.schema.json		
tipo	<i>objeto</i>	
propriedades		
• basic	<i>Básico</i>	
	tipo	<i>objeto</i>
	propriedades	
• username	<i>Nome de usuário</i>	
	tipo	<i>texto</i>
	exemplos	admin
	padrão	
• full_name	<i>Nome completo</i>	
	tipo	<i>texto</i>
	exemplos	Administrador do Weblate
	padrão	
• email	<i>E-mail</i>	
	tipo	<i>texto</i>
	exemplos	noreply@example.com
	padrão	
	formato	e-mail

continua na próxima página

Tabela 2 – continuação da página anterior

• profile	• date_joined	<i>Data de adesão</i>
	tipo	<i>texto</i>
	exemplos	2019-11-18T18:53:54.862Z
	padrão	
	formato	data-hora
	additionalProperties	False
	<i>Perfil</i>	
	tipo	<i>objeto</i>
	propriedades	
	• language	<i>Idioma</i>
	tipo	<i>texto</i>
	exemplos	cs
	pattern	^[^]*\$
	padrão	
	• suggested	<i>Quantidade de textos sugeridos</i>
	tipo	<i>inteiro</i>
	exemplos	1
	padrão	0
	• translated	<i>Quantidade de textos traduzidos</i>
	tipo	<i>inteiro</i>
	exemplos	24
	padrão	0
	• uploaded	<i>Quantidade de capturas de tela enviadas</i>
	tipo	<i>inteiro</i>
	exemplos	1
	padrão	0
	• hide_complete	<i>Ocultar as traduções concluídas no painel</i>
	tipo	<i>booleano</i>
	exemplos	False
	padrão	True
	• secondary_in_zen	<i>Mostra as traduções secundárias no modo Zen</i>
	tipo	<i>booleano</i>
	exemplos	True
	padrão	True
	• hide_source	<i>Ocultar a fonte caso uma tradução secundária exista</i>
	tipo	<i>booleano</i>
	exemplos	False
	padrão	True
	• editor_link	<i>Link do editor</i>
	tipo	<i>texto</i>
	exemplos	
	pattern	^.*\$
	padrão	
	• translate_mode	<i>Modo do editor de tradução</i>
	tipo	<i>inteiro</i>
	exemplos	0
	padrão	0
	• zen_mode	<i>Modo editor Zen</i>
	tipo	<i>inteiro</i>
	exemplos	0
	padrão	0
	• special_chars	<i>Caracteres especiais</i>
	tipo	<i>texto</i>
	exemplos	
	pattern	^.*\$

continua na próxima página

Tabela 2 – continuação da página anterior

• auditlog	• dashboard_view	padrão		
		<i>Visão padrão do painel</i>		
		tipo	<i>inteiro</i>	
		exemplos	1	
		padrão	0	
	• dashboard_component	<i>Lista de componentes padrão</i>		
		padrão	null	
		anyOf	tipo	<i>null</i>
	• languages	<i>Idiomas traduzidos</i>	tipo	<i>inteiro</i>
		tipo	<i>array</i>	
		padrão		
		itens	<i>Código do idioma</i>	
		tipo	<i>texto</i>	
		exemplos	cs	
		pattern	^.*\$	
		padrão		
	• secondary_language	<i>Idiomas secundários</i>		
		tipo	<i>array</i>	
		padrão		
		itens	<i>Código do idioma</i>	
		tipo	<i>texto</i>	
		exemplos	sk	
	• watched	<i>Projetos observados</i>	pattern	^.*\$
			padrão	
			tipo	<i>array</i>
			padrão	
			itens	<i>Slug do projeto</i>
• auditlog	• address	tipo	<i>texto</i>	
		exemplos	weblate	
		pattern	^.*\$	
		padrão		
	• user_agent	<i>Agente de usuário</i>	tipo	<i>texto</i>
			exemplos	PC / Linux / Firefox 70.0
			pattern	^.*\$
			padrão	
			tipo	<i>texto</i>
	• timestamp	<i>Carimbo de data e hora</i>	exemplos	2019-11-18T18:58:30.845Z
			tipo	<i>texto</i>
			exemplos	2019-11-18T18:58:30.845Z
			tipo	<i>texto</i>
			exemplos	2019-11-18T18:58:30.845Z
	• timestamp	<i>Carimbo de data e hora</i>	tipo	<i>texto</i>
			exemplos	2019-11-18T18:58:30.845Z
			tipo	<i>texto</i>
			exemplos	2019-11-18T18:58:30.845Z
			tipo	<i>texto</i>
	• timestamp	<i>Carimbo de data e hora</i>	exemplos	2019-11-18T18:58:30.845Z
			tipo	<i>texto</i>
			exemplos	2019-11-18T18:58:30.845Z
			tipo	<i>texto</i>
			exemplos	2019-11-18T18:58:30.845Z

continua na próxima página

Tabela 2 – continuação da página anterior

• activity	padrão	
	formato	data-hora
	Atividade	
	tipo	texto
	exemplos	login
	pattern	^.*\$
	padrão	
additionalProperties	False	
definições		

Ver também:

Perfil do usuário, *dumpuserdata*

3.11 Criando lançamento do Weblate

3.11.1 Agenda de lançamentos

O Weblate tem um ciclo de lançamento de dois meses para lançamentos (x.y). Geralmente, eles são seguidos por várias versões de correções de bugs para corrigir problemas que ocorrem neles (x.y.z).

A mudança na versão principal indica que o processo de atualização não pode ignorar esta versão - você sempre deve atualizar para x.0 antes de atualizar para versões x.y superiores.

Ver também:

Atualizando o Weblate

3.11.2 Planejamento de lançamento

Os recursos para os próximos lançamentos são coletados usando marcos do GitHub, você pode ver nosso roteiro em <https://github.com/WeblateOrg/weblate/milestones>.

3.11.3 Processo de lançamento

Coisas para verificar antes de fazer um lançamento:

1. Verificar novos idiomas traduzidos usando `./scripts/list-translated-languages`.
2. Definir a versão final usando `./scripts/prepare-release`.
3. Certificar-se de que as capturas de tela estejam atualizadas usando `make -j 12 -C docs update-screenshots`.
4. Junte todas as traduções pendentes usando `wlc push; git remote update; git merge origin/weblate`

Fazer o lançamento:

5. Criar um lançamento usando `./scripts/create-release --tag` (veja abaixo os requisitos).

Etapas manuais pós-lançamento:

6. Atualizar a imagem Docker.
7. Fechar o marco do GitHub.
8. Assim que a imagem Docker for testada, adicionar uma tag e fazer o push dela.

9. Atualizar um chart Helm para a nova versão.
10. Incluir a nova versão em `.github/workflows/migrations.yml` para cobri-la nos testes de migração.
11. Incrementar a versão nos links de download do site.
12. Incrementar a versão no repositório usando `./scripts/set-version`.
13. Verificar se o `readthedocs.org` compilou todas as traduções da documentação usando o `./scripts/rtd-projects`.

Para criar tags usando o script `./scripts/create-release`, você precisará do seguinte:

- GnuPG com chave privada usada para assinar o lançamento
- Acesso por push a repositórios git Weblate (ele envia tags)
- Ferramenta **hub** configurada e acesso para criar lançamentos no repositório Weblate
- Acesso SSH ao servidor de download Weblate (os downloads do site são copiados lá)

3.12 Segurança e privacidade

Dica: Na Weblate, a segurança mantém um ambiente que valoriza a privacidade de nossos usuários.

O desenvolvimento do Weblate segue as [Melhores Práticas da Core Infrastructure Initiative](#) da Linux Foundation.

Ver também:

Problemas de segurança

3.12.1 Security updates

Only the latest release is guaranteed to receive security updates.

3.12.2 Rastreamento de dependências para vulnerabilidades

Problemas de segurança em nossas dependências são monitorados usando [Dependabot](#). Isso cobre as bibliotecas Python e JavaScript, e a versão estável mais recente tem suas dependências atualizadas para evitar vulnerabilidades.

Dica: Pode haver vulnerabilidades em bibliotecas de terceiros que não afetam o Weblate, portanto, não são solucionadas com o lançamento de versões de correção de bugs do Weblate.

3.12.3 Segurança de contêiner Docker

Os contêineres Docker são verificados regularmente usando [Anchore](#) e [Trivy](#).

Isso nos permite detectar vulnerabilidades antecipadamente e lançar melhorias rapidamente.

Você pode obter os resultados dessas varreduras no GitHub – eles são armazenados como artefatos em nosso CI no formato SARIF (Static Analysis Results Interchange Format).

Ver também:

Integração contínua

3.13 Contribuindo para módulos do Weblate

Além do repositório principal, o Weblate consiste em vários módulos Python. Todos estes seguem a mesma estrutura e esta documentação abrange todos eles.

Por exemplo, isso cobre:

- `wlc`, biblioteca Python cliente, veja [Weblate Client](#)
- `translation-finder`, usado para descobrir arquivos traduzíveis no repositório
- `language-data`, definições de idiomas para o Weblate, veja [Definições de idioma](#)

3.13.1 Diretrizes de codificação

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 usuário.

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 Executando testes

Os testes são executados usando `py.test`. Primeiro você precisa instalar os requisitos de teste:

```
pip install -r requirements-test.txt
```

Você pode então executar o test suite no checkout do repositório:

```
py.test
```

Ver 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 usando o formatador de código **black**.

Para verificar a qualidade do código, você pode usar **flake8**, os plugins recomendados estão listados em `.pre-commit-config.yaml` e sua configuração é colocada em `setup.cfg`.

A abordagem mais fácil para impor tudo isso é instalar [pre-commit](#). O repositório contém a configuração para verificar se os arquivos do commit estão sãos. Depois de instalá-lo (ele já está incluído no `requirements-lint.txt`), ative-o executando `pre-commit install` na sua cópia do Weblate. Desta forma, todas as suas alterações serão verificadas automaticamente.

Você também pode acionar a verificação manualmente, para verificar todos os arquivos executados:

```
pre-commit run --all
```

Ver também:

[Código-fonte do Weblate](#)

3.14 Sobre o Weblate

3.14.1 Objetivos do projeto

Ferramenta de localização contínua baseada na web com *Integração com controle de versão* suportando uma ampla gama de *formatos de arquivos*, facilitando a contribuição dos tradutores.

3.14.2 Nome do projeto

“Weblate” é uma palavra-valise das palavras “web” e “translate”.

3.14.3 Site 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 contatado em michal@weblate.org.

3.14.6 Autores

Weblate foi iniciado por Michal Čihař. Desde sua criação, em 2012, milhares de pessoas contribuíram.

3.15 Licença

Informações de licenciamento mais detalhadas estão disponíveis no código-fonte Weblate e seguem a especificação REUSE 3.0 <<https://reuse.software/>>`_.

Direitos autorais © Michal Čihař michal@weblate.org

Este programa é um software livre: você pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU, conforme publicado pela Free Software Foundation, seja a versão 3 da Licença, ou (a seu critério) qualquer versão posterior.

Este programa é distribuído na esperança de que ele seja útil, mas sem qualquer garantia; sem sequer a garantia implícita de COMERCIALIZAÇÃO ou ADEQUAÇÃO PARA UM PROPÓSITO ESPECÍFICO. Consulte a Licença Pública Geral GNU para obter mais detalhes.

Você deve ter recebido uma cópia da Licença Pública Geral GNU junto com este programa. Caso contrário, veja <<https://www.gnu.org/licenses/>>.

Histórico de alterações

4.1 Weblate 4.16

Released on March 1st 2023.

- Format string checks now also detects duplicated formats.
- Improved search performance for some specially formatted strings.
- Celery beat is now storing the tasks schedule in the database.
- Added support for IBM Watson Language Translator.
- Dropped support for VCS integration settings deprecated in 4.14.
- Added support for Bitbucket Server pull requests.
- Improved conflicts handling in gettext PO files.
- Added support for defining strings state when adding via API.
- Added support for configuring CORS allowed origins.
- Added plurals support to automatic suggestions.

[All changes in detail.](#)

4.2 Weblate 4.15.2

Released on January 25th 2023.

- Enabled gettext JSON and i18next v4 formats in the default configuration.
- Fixed crash on uploading corrupted files.
- Show stale directories in Git repository status.

[All changes in detail.](#)

4.3 Weblate 4.15.1

Released on January 19th 2023.

- Sugestões corrigidas da tradução automática.
- Corrigida a falha da página de complemento em alguns casos específicos.
- Corrigido modelo sem tradução para novas traduções em alguns casos.
- Licenciamento documentado usando [REUSE 3.0](#).
- Paginação fixa dos usuários na gestão da equipe.
- Melhorado o desempenho de criação e salvamento de projetos.
- Adicionado suporte para arquivos JSON gotext.
- Adicionado suporte para arquivos i18next v4.
- A paginação na API agora é personalizável.

[Todas as alterações em detalhes.](#)

4.4 Weblate 4.15

Lançado em 16 de dezembro de 2022.

- Adicionado suporte para alterações de navegação para um texto individual.
- Corrigido o tratamento de plurais na tradução automática de outros componentes.
- Adicionado atalho de teclado Alt+Enter para enviar texto como sugestão.
- Adicionado suporte para objetos colocáveis no formato Fluent.
- Melhorado o desempenho da memória de tradução.
- Gera automaticamente links de navegação do repoweb para serviços de hospedagem de código conhecidos.
- Melhorado o desempenho de várias visões.
- Melhorada a listagem de textos com plurais.
- Adicionado suporte para adicionar marcação personalizada ao cabeçalho HTML.
- Corrigida a geração de arquivos MO na extensão para incluir apenas arquivos traduzidos.
- Corrigida a renderização de marcadores de expressão regular.
- Melhorado o comportamento de verificação de espaços reservados com plurais.
- Adicionado suporte para nomenclatura de arquivos de tradução adequados para Google Play.
- Adicionado suporte para etiquetas em API.
- Adicionado suporte para escolher e-mail de commits diferente do de notificações.
- A imagem do Docker não habilita mais o modo de depuração por padrão.
- Ordena os termos do glossário com base na prioridade do componente do glossário.
- Adicionados administradores de equipe que podem adicionar ou remover membros da equipe.
- Adicionada um pop-up de confirmação antes de excluir usuários.
- Adicionado add-on para personalizar a saída XML.

[Todas as alterações em detalhes.](#)

4.5 Weblate 4.14.2

Lançado em 5 de novembro de 2022.

- Adicionado o suporte a remover entradas da memória de tradução.
- Melhorada a análise no alerta de idioma duplicado.
- Melhorada a precisão de verificação de palavras consecutivas duplicadas.
- Melhorado o escalonamento de enviar muitas notificações.
- Melhorado tratamento de estado do texto para tradução de legendas.
- Descontinuada a configuração insegura de chaves de API do serviço VCS por meio da configuração `_TOKEN/_USERNAME` em vez da lista `_CREDENTIALS`.
- Corrigido o processamento de alguns arquivos CSV enviados.
- Melhorado o tratamento de alterações de espaços em branco na exibição de comparação.
- Adicionado link de gerenciamento de sugestões automáticas para páginas de gerenciamento.
- Acompanha a remoção/resolução de comentários no histórico.
- Corrigida a restauração de backups de projetos com componentes vinculados.
- Corrigida a entrada de captcha no registro malsucedido.
- Melhorado o suporte a idiomas no DeepL.
- Melhorada a compatibilidade dos webhooks com repositórios autenticados.
- Suporte adicionado para Python 3.11.

Todas as alterações em [detalhes](#).

4.6 Weblate 4.14.1

Lançado em 15 de setembro de 2022.

- Corrigida a geração de backups d2 projeto em algumas situações.
- Melhorado o relatório de erro no envio de arquivo.
- Obtém todos os e-mails, verificados pelo usuário, do GitHub durante a autenticação.
- Evita corresponder os termos de glossário no contexto ou nas chaves.
- Adicionadas notificações para remoções de texto.
- Melhorado o gerenciamento de termos não traduzidos no glossário.
- Lista o número de membros da equipe na página de gerenciamento da equipe.
- Adiciona interface de gerenciamento de grupos.
- Sempre mostra as estatísticas de revisão quando revisões estão habilitadas.
- Adicionado suporte de pesquisa em API de unidades.
- Corrigida a exibição da barra de progresso para textos somente leitura no fluxo de trabalho de revisão.
- Melhorada a verificação de pontuação de Birmanês.
- Corrigida a coleta de lixo de dados de métricas.

Todas as alterações em [detalhes](#).

4.7 Weblate 4.14

Lançado em 22 de agosto de 2022.

- Rastreia as alterações de extensões em um histórico.
- Corrigida a análise da tradução de arquivos do Windows RC, HTML e texto.
- Estendidas as opções de configuração de estilo de código de idioma.
- Adicionado suporte para plurais atualizado nas versões recentes do CLDR.
- Reduzido o uso de memória durante a atualização de componentes com muitas traduções.
- Adicionado suporte para domínio de tradução em SAP Translation Hub.
- Permite links absolutos em locais de texto fonte.
- Melhorada a operação por trás de alguns proxies reversos.
- Estendida a API para cobrir a memória de tradução.
- Melhorado o fluxo de tradução de documentos.
- Melhorada a confiabilidade da tradução de arquivos HTML e texto.
- Adicionado suporte para backups em nível de projeto.
- Melhorado o desempenho e uso de memória de pesquisas de memória de tradução.

[Todas as alterações em detalhes.](#)

4.8 Weblate 4.13.1

Lançado em 1º de julho de 2022.

- Corrigidas rastreamento de sugestões no histórico.
- Corrigida a análise de informações de proxy reverso do Cloudflare.
- Faz com que um erro de análise bloqueie a tradução de um componente.
- Corrigida a configuração do arquivo intermediário na extensão de descoberta.
- Corrigido o comportamento das traduções do DeepL com espaços reservados.
- Corrigida a remoção de tradução de textos por meio de API.
- Suporte adicionado para remover usuário de um grupo por meio de API.
- Corrigido o log de auditoria para e-mails de convite de usuário.
- Corrigidos nomes de marcadores para textos de formatação Java.

[Todas as alterações em detalhes.](#)

4.9 Weblate 4.13

Lançado em 15 de junho de 2022.

- Alterado o comportamento de atualização de nomes de idiomas.
- Adicionada paginação à listagem de projetos.
- A API para criar novas unidades agora retorna informações sobre a unidade recém-criada.
- A descoberta de componentes agora oferece suporte à configuração de um idioma intermediário.

- Adicionadas variantes de codificação fixa aos formatos CSV.
- Alterado o tratamento de contexto e local para alguns formatos para melhor adequação à implementação subjacente.
- Adicionado suporte para o formato ResourceDictionary.
- Melhoradas as cores da barra de progresso para daltônicos.
- Corrigida a limpeza de variantes na remoção de texto.
- Compatibilidade com Django 4.1.
- Adicionado suporte para armazenar elementos XML com escape em XLIFF.
- Melhorada a formatação de erros de verificação de espaço reservado.
- Redireciona /.well-known/change-password para /accounts/password/.
- Os serviços de tradução de máquina agora são configuráveis por projeto.
- Adicionada uma permissão separada para resolver comentários e concedê-la à função *Revisar textos*.
- Adicionado suporte para armazenar traduções alternativas no arquivo CSV.
- A verificação de espaços reservados agora também não diferencia maiúsculas de minúsculas.

[Todas as alterações em detalhes.](#)

4.10 Weblate 4.12.2

Lançado em 11 de maio de 2022.

- Corrigida a reconstrução da memória de tradução do projeto para alguns componentes.
- Corrigida a classificação de componentes por textos não traduzidos.
- Corrigida uma possível perda de traduções ao adicionar novo idioma.
- Garante que a chave SSH do Weblate seja gerada durante as migrações.

[Todas as alterações em detalhes.](#)

4.11 Weblate 4.12.1

Lançado em 29 de abril de 2022.

- Corrigido título da mensagem de pull requests.
- Melhorado o tratamento de erros de sintaxe no formato Fluent.
- Corrigida a exibição de avatares em e-mails de notificação.
- Adiciona suporte para monetização na web.
- Corrigida a remoção de textos fonte obsoletos ao remover traduções.

[Todas as alterações em detalhes.](#)

4.12 Weblate 4.12

Lançado em 20 de abril de 2022.

- Adicionado suporte para Amárico em *Ponto final não correspondente*.
- Adicionado suporte para birmanês em *Ponto de interrogação não correspondente*.
- Extendidas as opções da extensão *Geração de pseudolocalidade*.
- Adicionado o marcador `ignore-all-checks` para ignorar todas as verificações de qualidade em um texto.
- Evita que a extensão *Geração de pseudolocalidade* acione verificações com falha.
- Adicionado suporte para *Pull requests do Gitea*.
- Adicionado código de idioma de estilo Linux a *Estilo de código de idioma*.
- Adicionado o suporte a recompilar a memória de tradução de projeto.
- Melhorada a API para criar componentes de um arquivo.
- Adiciona botões de cópia e clonagem a outras traduções.
- Torne a mensagem de merge request configurável a nível do componente.
- Melhorado o comportamento de restrição de comprimento máximo com tags XML.
- Corrigido o carregamento de arquivos Fluent com comentários adicionais.

Todas as alterações em detalhes.

4.13 Weblate 4.11.2

Lançado em 4 de março de 2022.

- Corrigidos arquivos MO corrompidos no lançamento binário.

Todas as alterações em detalhes.

4.14 Weblate 4.11.1

Lançado em 4 de março de 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) <<https://github.com/WeblateOrg/weblate/security/advisories/GHSA-3872-f48p-pxqj>> para mais detalhes.
- Corrigido o carregamento de textos aproximados de arquivos CSV.
- Adicionado suporte para criar equipes usando a API.
- Corrigida a menção de usuários na exibição de sugestões.
- O acesso com token de projetos agora podem ser personalizados.

Todas as alterações em detalhes.

4.15 Weblate 4.11

Lançado em 25 de fevereiro de 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*.
- Indica o bloqueio e o acordo do colaborador na lista de outras ocorrências.
- Corrigida a atualização de arquivos PO com textos obsoletos ou plurais ausentes.
- Aprimorada a compatibilidade da extensão de squash com Gerrit.
- Inicializa automaticamente os idiomas do usuário com base no cabeçalho *Accept-Language*.
- Melhorado o tratamento de erros na remoção de texto.
- O Weblate agora requer Python 3.7 ou mais recente.
- Corrigidas algumas operações de gravação com autenticação de token do projeto.
- Corrigido o rastreamento do estado de textos quando os textos são alterados no repositório.
- Rastreia as alterações de texto do repositório.
- Cabeçalho fixo na listagem de traduções para melhorar a navegação.
- Corrigida a não tradução de textos em *Propriedades Java*.
- Corrigida a operação do Git com nomes de ramos não ASCII.
- Nova extensão *Preencher tradução com fonte*.
- Adicionado o *Estilo de mesclagem Mesclar sem fast-forward*.
- Corrigido o acionamento da extensão *Tradução automática* em textos recém-adicionados.
- Melhoradas as verificações de pontuação para Birmanês.
- Adicionado suporte para definir equipes personalizadas a nível de projeto para conceder acesso a usuários, veja *Gerenciando controle de acesso por projeto*.
- Adicionado links da documentação para alertas.
- O contêiner Docker habilita automaticamente TLS/SSL para e-mail de saída quando necessário.
- Adicionado suporte para pesquisar para comentários resolvidos.
- Suporte adicionado para borgbackup 1.2.
- Corrigida a aplicação do rótulo *Traduzido automaticamente*.

[Todas as alterações em detalhes.](#)

4.16 Weblate 4.10.1

Lançado em 22 de dezembro de 2021.

- Documentadas as alterações introduzidas pela atualização para Django 4.0.
- Corrigida a exibição do rótulo *Traduzido automaticamente*.
- Corrigida a exibição de API de 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 usada*.
- O contêiner Docker do Weblate agora usa Python 3.10.

Todas as alterações em detalhes.

4.17 Weblate 4.10

Lançado em 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” não é mais parte do grupo *Idiomas*.
- “Efetuar tradução automática” está em *Administração* e no novo grupo “Tradução automática”.
- Corrigida a geração de arquivos XLSX em caracteres especiais.
- Adicionada capacidade ao backend de autenticação do GitHub para verificar se o usuário pertence a uma organização ou equipe específica do GitHub.
- Melhorado o feedback ao serem passados parâmetros inválidos para a API.
- Adicionado suporte para tokens de acesso com escopo de projeto para a API.
- Corrigida a remoção de texto em alguns casos.
- Corrigida a tradução de textos recém-adicionados.
- Etiqueta os textos traduzidos automaticamente para facilitar sua filtragem.

Todas as alterações em detalhes.

4.18 Weblate 4.9.1

Lançado em 19 de novembro de 2021.

- Corrigido o envio de arquivos 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.

4.19 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 arquivo 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 aplicativos 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 arquivos MO para traduções monolíngues.
- Adicionado endpoint de API para baixar todas as traduções de componentes como um arquivo ZIP.
- Suporte adicionado para Python 3.10.
- Adicionado suporte para reenviar convite por e-mail a partir da interface de gerenciamento.

Todas as alterações em detalhes.

4.20 Weblate 4.8.1

Lançado em 10 de setembro de 2021.

- Corrigida a remoção de usuário na interface administrativa do Django.
- Documentados os parâmetros de extensões com mais detalhes.
- Corrigido um erro de JavaScript no glossário.
- Adicionado limite ao número de correspondências na verificação de consistência.
- Melhorado o manuseio de espaços reservados em traduções de máquina.
- Corrigida a criação de extensões usando 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.

4.21 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 arquivo para downloads.
- Adicionado suporte para o formato Fluent.
- Adicionado suporte para usar tabulações para formatos de recuo do JSON.

[Todas as alterações em detalhes.](#)

4.22 Weblate 4.7.2

Lançado em 15 de julho de 2021.

- Suporta mais apelidos de idioma a serem configurados em um 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 arquivos RC do Windows.
- Corrigida possível falha na atualização de XLIFF.

[Todas as alterações em detalhes.](#)

4.23 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 arquivo.

[Todas as alterações em detalhes.](#)

4.24 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 de Object Pascal*.
- Renomeado *Chaves próximas* para *Chaves similares* para melhor descrever o propósito.
- Adicionado suporte para *Arquivos 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 usuários quando a autenticação por e-mail está desabilitada.
- Melhoradas as definições de idioma.
- Adicionado suporte para bloquear usuários de contribuir para um projeto.
- Corrigida a criação automática de idiomas de glossário.
- Estendida a documentação sobre extensões.
- Melhorias no desempenho para componentes com repositórios vinculados.
- Adicionado suporte para DeepL API grátis.
- O usuário de gerência não precisa mais da interface administrativa do Django.

Todas as alterações em detalhes.

4.25 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 arquivos de propriedades vazios.
- Corrigido o alinhamento de ícone de cópia em idiomas RTL.
- Estendidas as estatísticas de texto na aba de Informações.
- Corrigido o tratamento de arquivos 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.

4.26 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 usuário 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](#).

4.27 Weblate 4.6

Lançado em 19 de abril de 2021.

- O comando de gerenciamento `auto_translate` agora tem um parâmetro para especificar o modo de tradução.
- Adicionado suporte para *Arquivos texto*.
- Adicionadas tendências e métricas para todos os objetos.
- Adicionado suporte para copiar diretamente o texto a partir de idiomas secundários.
- 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 superusuários.
- Melhorada a desempenho e confiabilidade da extensão de tradução automática.
- A limitação de taxa agora pode ser personalizada no contêiner do Docker.
- API para criar componentes agora usa automaticamente *URLs internas do Weblate*.
- Indicação de estado simplificada ao listar textos.
- Hash de senhas agora usa Argon2 por padrão.
- Barras de progresso simplificadas indicando o status da tradução.
- Renomeada *Adição de idiomas em falta* para esclarecer seu propósito.
- Corrigido o salvamento do estado de texto para XLIFF.
- Adicionada pesquisa para todo idioma.
- Suporte inicial a *Dimensionamento horizontal* da implantação do Docker.

Todas as alterações em [detalhes](#).

4.28 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.

4.29 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 *Há palavras duplicadas de forma consecutiva*.
- Permitido o envio de uma captura de tela de uma página de tradução.
- Adicionada sincronização de arquivo 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.

4.30 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 arquivos TMX.
- Melhor manuseio para envios de PO monolíngues usando API.
- Melhorada o comportamento de alertas em componentes de glossário.
- Melhoradas verificações de links 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 arquivos ODF.
- Interface melhorada para personalização de downloads de arquivos de tradução

[Todas as alterações em detalhes.](#)

4.31 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 arquivos TermBase eXchange.
- Adicionado suporte para definir manualmente variantes de texto usando 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.
- Habilita 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 em um 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 usuário 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.](#)

4.32 Weblate 4.4.2

Lançado em 14 de janeiro de 2021.

- Corrigida a corrupção de um arquivo MO distribuído.

4.33 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 usuários.
- Melhorado o tratamento de contexto em arquivos 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 arquivos CSV.
- Uso de compactação de conteúdo para downloads de arquivos.
- Melhorada a experiência do usuário na importação de arquivo ZIP.
- Melhorada a detecção do formato de arquivo 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.](#)

4.34 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 gerenciamento.
- Corrigido o tratamento de estado somente leitura na edição em massa.
- Integração com CodeMirror melhorada.
- Adicionada extensão para remover textos vazios de arquivos 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 arquivo de tradução para o idioma fonte.
- O usuário 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 aba “outras traduções” ao traduzir.
- Adicionada API de tarefas.

- Melhorado o desempenho do envio de arquivo.
- Melhorada a exibição de caracteres especiais definidos pelo usuário.
- Melhorado o desempenho da tradução automática.
- Várias pequenas melhorias na interface do usuário.
- Melhorada a nomenclatura de downloads ZIP.
- Adicionada a opção para obter notificações de projetos não observados.

[Todas as alterações em detalhes.](#)

4.35 Weblate 4.3.2

Lançado em 4 de novembro de 2020.

- Corrigida a falha em algumas máscaras de arquivo de componentes.
- Melhorada a precisão de verificação de palavras consecutivas duplicadas.
- Adicionado suporte a pull requests do Pagure.
- Melhoradas as mensagens de erro para registros com falha.
- Revertida a renderização de comentários do desenvolvedor 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 seu significado.
- As pull requests do GitLab em repositórios com caminhos mais longos.

[Todas as alterações em detalhes.](#)

4.36 Weblate 4.3.1

Lançado em 21 de outubro de 2020.

- Melhorada o desempenho da tradução automática.
- Expiração de sessão fixa para usuários autenticados.
- Adicionar suporte para ocultar informação de versão.
- Melhorada 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 usuário de um projeto.

[Todas as alterações em detalhes.](#)

4.37 Weblate 4.3

Lançado em 15 de outubro de 2020.

- Incluídas estatísticas de usuário 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.
- Corrigidas a contagem das estatísticas após remoção de sugestão.
- Estendido perfil público de usuário.
- Corrigida a configuração de verificações reforçadas.
- Melhorada a documentação sobre backups embutidos.
- Movido o atributo do idioma fonte do projeto para um componente.
- Adicionada verificação de formatação Vue I18n.
- A verificação de espaços reservados genéricos agora oferece suporte a expressões regulares.
- Melhorado o visual do modo Matriz.
- Maquinaria agora é chamada de 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 arquivos ZIP ou envio de documentos.
- Códigos de status de resposta API consolidados.
- Suporte a Markdown no acordo de colaborador.
- Melhorado o rastreamento de textos fontes.
- Melhorada a compatibilidade dos formatos JSON, YAML e CSV.
- Suporte adicionado para remover Fios da Meada.
- Melhorado o desempenho de downloads de arquivos.
- Melhorada a visão de gerenciamento de repositório.
- Habilita automaticamente o formato java para Android.
- Suporte adicionado para prints da tela localizados.
- Suporte adicionado para Python 3.9.
- Corrigida a tradução de arquivos HTML sob certas condições.

[Todas as alterações em detalhes.](#)

4.38 Weblate 4.2.2

Lançado em 2 de setembro de 2020.

- Corrigido correspondência de textos fonte para formatos JSON.
- Corrigido o redirecionamento de login para algumas configurações de autenticação.
- Corrigida autenticação por LDAP com sincronização de grupo.
- Corrigida falha em relatar progresso das traduções automáticas.
- Corrigido o “squashing” de commits do Git com linhas finais habilitadas.
- Corrigida a criação de componentes VCS locais usando API.

4.39 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 arquivos XLIFF.
- Permitida a configuração de CDN de localização na imagem Docker.

4.40 Weblate 4.2

Lançado em 18 de agosto de 2020.

- Melhoradas as páginas de usuário e adicionada uma listagem de usuários.
- 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.
- O número de textos próximos exibidos pode ser configurado.
- Adicionar suporte para bloquear componentes que sofrem erros de 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 usuário 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 arquivo.
- Removido o suporte para alguns redirecionamentos de URL de compatibilidade.
- Adicionada verificação para literais de modelo de ECMAScript.
- Adicionada opção para observar um componente.

- Removido o ponto inicial das chaves da unidade JSON.
- Removida fila separada de Celery para memória de tradução.
- Permite traduzir todos os componentes um idioma 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 do JavaScript](#).
- O domínio Weblate é agora configurado nas configurações, veja [SITE_DOMAIN](#).
- Adiciona suporte para pesquisar componente e projeto.

4.41 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 arquivos 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](#).

4.42 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 textos fonte com captura de tela.
- 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 usuário em alguns casos.
- Permitida a adição à lista de permissão métodos de registro com registros fechados.
- Melhorada a pesquisa de termos relacionados no glossário.
- Melhoradas as correspondências de memória de tradução.
- Agrupados os mesmos resultados de maquinaria.
- Adicionado um link direto para editar captura de tela 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 arquivo: *Textos do PHP de Laravel*, *Arquivos HTML*, *Formato OpenDocument*, *Formato IDML*, *Arquivos RC do Windows*, *Traduções de INI*, *Traduções de Inno Setup INI*, *Propriedades GWT*, *Arquivos go-i18n de JSON*, *Arquivo ARB*.
- Uso consistentemente de dispensado como um estado de verificações dispensadas.
- Adiciona suporte para configurar extensões padrão para habilitar.
- 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 usuário para facilitar o seu início.
- Melhorada a análise de código de idioma.
- Mostra as traduções no idioma do usuário 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*, *Há palavras duplicadas de forma consecutiva*.
- 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 marcadores de tradução, veja *Personalizando comportamento usando marcadores*.
- Sempre mostra a verificação de texto renderizado se habilitada.
- A API agora tem suporte à filtragem de alterações.
- Adicionado suporte para compartilhar glossários entre projetos.

4.43 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 usando 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 tela.
- Corrigidas notificações de resumo mensais.
- Corrigido o comportamento de tradução intermediário com unidades não existentes na tradução.

4.44 Weblate 4.0.3

Lançado em 2 de maio de 2020.

- Corrigida possível falha em relatórios.
- As menções do usuário nos comentários agora não diferenciam maiúsculas de minúsculas.
- Corrigida a migração do PostgreSQL para não superusuários.
- Corrigida a alteração da URL do repositório durante a criação do componente.
- Corrigida uma falha quando o repositório upstream desaparece.

4.45 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.
- Melhorado o 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.

4.46 Weblate 4.0.1

Lançado em 16 de abril de 2020.

- Corrigida a instalação de pacotes do PyPI.

4.47 Weblate 4.0

Lançado em 16 de abril de 2020.

- O Weblate agora requer Python 3.6 ou mais recente.
- Adicionada visão geral do gerenciamento de alertas de componentes.
- Adicionado alerta de componente para URLs quebradas do navegador do repositório.
- Melhoradas páginas de autenticação e registro.
- 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 porcentagem.
- Exibe verificações com falha em sugestões.
- Registra as alterações de texto fonte no histórico.
- Atualizado o Microsoft Translator para a versão 3 da API.
- Reimplementado backend da memória de tradução.
- Adicionado suporte a várias pesquisas `is:` em *Pesquisando*.
- Permite fazer *Tradução não alterada* evitar lista negra interna.
- Melhorada a extração de comentários de arquivos po monolíngues.
- Renomeadas as mensagens do quadro de comunicação para anúncios.
- Corrigidos problemas ocasionais com e-mails de registro.
- Melhorada a extensão de atualização do LINGUAS para lidar com mais variantes de sintaxe.
- Corrigida a edição do arquivo fonte XLIFF monolíngue.
- Adicionado suporte para correspondência exata em *Pesquisando*.
- Estendida a API para cobrir capturas de tela, usuários, grupos, listas de componentes e projetos de criação estendida.
- Adicionado suporte para envio de fonte em traduções bilíngues.
- Adicionado suporte para idioma intermediário de desenvolvedores.
- Adicionado suporte para revisão de textos fonte.
- Estendidas as opções de download para memória de tradução em toda a plataforma.

4.48 Série 3.x do Weblate

4.48.1 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 links para eventos nas notificações de resumo.

- Corrigido o e-mail para confirmação de remoção de conta.
- Adicionado suporte para autenticação Slack no contêiner 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 porcentagem.
- Corrigida a autenticação usando alguns provedores de terceiros.

4.48.2 Weblate 3.11.2

Lançado em 22 de fevereiro de 2020.

- Corrigida a renderização de sugestões.
- Corrigidos alguns textos relatados incorretamente como não tendo palavras.

4.48.3 Weblate 3.11.1

Lançado em 20 de fevereiro de 2020.

- Documentadas as alterações de configuração no Celery.
- Melhorada a validação de nome de arquivo 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.

4.48.4 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.
- Corrigidos os links 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.
- Adicionado suporte para agrupar formas de textos.
- 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.
- Corrigido o download de textos que necessitam ação.
- Novo alerta para alertar sobre o uso da mesma máscara de arquivo 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 *Forçando verificações*.
- Aumento da validade padrão dos links 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.

4.48.5 Weblate 3.10.3

Lançado em 18 de janeiro de 2020.

- Suporte para translate-toolkit 2.5.0.

4.48.6 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.

4.48.7 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 contêiner Docker.

4.48.8 Weblate 3.10

Lançado em 20 de dezembro de 2019.

- Melhorada a interface do usuário da aplicação.
- 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.

4.48.9 Weblate 3.9.1

Lançado em 28 de outubro de 2019.

- Remover alguns arquivos 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 registro.
- Corrigido o limitador de taxa de acerto de remoção de conta.
- Adicionada pesquisa baseada em prioridade.
- Consertada possível falha ao adicionar textos para um arquivo JSON.
- A verificação e correção de HTML seguro agora respeitam a marcação do texto fonte.
- Impedir o envio de notificações para usuários convidados e deletados.
- Consertar a conexão SSL para o redis no Celery num contêiner Docker.

4.48.10 Weblate 3.9

Lançado em 15 de outubro de 2019.

- Incluir metadados do Weblate em arquivos baixados.
- 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 o download ZIP para mais arquivos 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 aplicativos 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 usando BorgBackup.
- Divida a extensão de limpeza para RESX para evitar atualizações indesejadas de arquivos.
- Adicionados recursos de pesquisa avançada.
- Permitido que os usuários baixem seus 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.

4.48.11 Weblate 3.8

Lançado em 15 de agosto de 2019.

- Adicionado suporte para criação simplificada de componentes semelhantes.
- Adicionado suporte para analisar marcadores de tradução de formatos de arquivo baseados em XML.
- Registra 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 senha.
- 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 arquivo 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 tela animadas.
- Melhorado o manuseio de arquivos 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 um link 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 gerenciamento para administradores.
- A prioridade do texto agora pode ser especificada usando marcadores.
- Adicionadas visões de gerenciamento 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.

4.48.12 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 usuário.
- Melhorado o tratamento de arquivos de tradução não suportados, mas descobertos.
- Relata mais detalhadamente os requisitos de formato de arquivo ausentes.

4.48.13 Weblate 3.7

Lançado em 21 de junho de 2019.

- Adicionada fila Celery separada para notificações.
- Usa uma aparência consistente com o aplicativo para 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 arquivo.
- Usa modelos para link 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 registro de erros.
- Permite local personalizado na extensão de geração de MO.
- Adicionadas extensões para limpar sugestões ou comentários antigos.
- Adicionada a opção para habilitar 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 gerenciamento 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 usuário ao mostrar o status da tradução.
- Novo logotipo e esquema de cores do Weblate.
- Nova aparência de emblemas de bitmap.

4.48.14 Weblate 3.6.1

Lançado em 26 de abril de 2019.

- Melhorado o manuseio de arquivos XLIFF monolíngues.
- Corrigidas as notificações de resumo em alguns casos complexos.
- Corrigido um alerta de erro de script de extensão.
- Corrigido o arquivo de geração de MO para arquivos 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.

4.48.15 Weblate 3.6

Lançado em 20 de abril de 2019.

- Adiciona suporte para baixar dados do usuário.
- Extensões agora são acionados automaticamente após a instalação.
- Melhoradas as instruções para resolver conflitos de mesclagem.
- A extensão de limpeza agora é compatível com traduções de metadados da loja de aplicativos.
- 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 usuários.
- Corrigida a descrição de alternar desabilitando a manipulação do cabeçalho `Language-Team`.
- Notifica os usuários mencionados nos comentários.
- Removida a detecção automática do formato de arquivo da configuração do componente.
- Corrigido o arquivo de geração de MO para arquivos 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.

4.48.16 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 arquivo 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.

4.48.17 Weblate 3.5

Lançado em 3 de março de 2019.

- Melhorado o desempenho da memória de tradução embutida.
- Adicionada interface para gerenciar a memória de tradução global.
- Melhorado o alerta sobre estado inválido do componente.
- Adicionada interface de usuário para gerenciar mensagens do quadro de comunicações.
- A mensagem de commit da extensão agora pode ser configurada.
- Reduz o número de commit 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 arquivos de metadados de loja de aplicativos.
- 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 arquivo XLSX.
- Compatibilidade com Tesseract 4.0.
- A extensão de cobrança agora remove projetos para cobranças não pagas após 45 dias.

4.48.18 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 arquivos de tradução.
- Mensagens de commit de mesclagem personalizáveis para Git.
- Adicionada indicação visual de alertas de componentes na navegação.
- Melhorado o desempenho de carregamento de arquivos 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 arquivo para oferecer suporte a mais parâmetros.

4.48.19 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 arquivos que não podem ser escritos dentro do DATA_DIR.
- Melhorada a exportação de CSV para alterações.

4.48.20 Weblate 3.2.2

Lançado em 20 de outubro de 2018.

- Remove a dependência Babel que não é mais necessária.
- Definições de idioma atualizadas.
- Melhora a documentação para extensões, LDAP e Celery.
- Corrigida a ativação de novos marcadores dos-eol e auto-java-messageformat.
- Corrigido o teste setup.py em execução do pacote PyPI.
- Melhorado o tratamento de plurais.
- Corrigida a falha da API de envio de tradução em alguns casos complexos.
- Corrigida a atualização da configuração do Git caso ela fosse alterada manualmente.

4.48.21 Weblate 3.2.1

Lançado em 10 de outubro de 2018.

- Documenta a dependência de backports.csv no Python 2.7.
- Corrige os testes em execução como root.
- Melhorado o tratamento de erros no módulo gitexport.
- Corrigido o relatório de progresso para idiomas recém-adicionados.
- Comunica corretamente os erros do worker do Celery ao Sentry.
- Corrigida a criação de novas traduções com o Qt Linguist.
- Corrigidas falhas ocasionais de atualização de índice de texto completo.
- Melhorada a validação ao criar novos componentes.
- Adicionado suporte para limpeza de sugestões antigas.

4.48.22 Weblate 3.2

Lançado em 6 de outubro de 2018.

- Adiciona o comando de gerenciamento `install_addon` para instalação de extensão automatizada.
- Permite configurações de limite de proporção mais refinadas.
- Adicionado suporte para exportação e importação de arquivos Excel.
- Melhora a limpeza de componentes no caso de complementos de descoberta de vários 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 usuários.
- Memória de tradução embutida agora armazena automaticamente as traduções feitas.
- Usuários e projetos podem importar suas memórias de tradução existentes.
- Melhor gerenciamento de textos relacionados às capturas de tela.
- Adicionado suporte para verificar Java MessageFormat.

Vea o [marco 3.2 no GitHub](#) para uma lista detalhada dos problemas resolvidos.

4.48.23 Weblate 3.1.1

Lançado em 27 de julho de 2018.

- Corrigida a falha do conjunto de teste em algumas configurações.

4.48.24 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 libavatar que será encerrado em breve.
- Nova extensão para marcar traduções não alteradas como necessitando de edição.
- Adiciona suporte para pular para um local específico durante a tradução.
- As traduções baixadas agora podem ser personalizadas.
- Melhorado o cálculo de similaridade de texto em correspondências de memória de tradução.
- Adicionado suporte para assinar commits do Git pelo GnuPG.

4.48.25 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.

4.48.26 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 gerenciamento `import_project` agora tem parâmetros ligeiramente diferentes.
- Adicionado suporte básico para arquivos Windows RC.
- Nova extensão para armazenar nomes de contribuidores em cabeçalhos de arquivos 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 em um projeto tenham as mesmas traduções.
- Suporte para mais variáveis em modelos de mensagem de commit.
- Adiciona suporte para fornecer contexto textual adicional.

4.49 Série 2.x do Weblate

4.49.1 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 baixar apenas textos que necessitam 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.

4.49.2 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 arquivo.

4.49.3 Weblate 2.19

Lançado em 15 de fevereiro de 2018.

- Corrigidas as importações em alguns formatos de arquivo.
- Exibe informações de navegador amigáveis para humanos no registro de auditoria.
- Adicionado um exportador TMX para arquivos.
- Várias melhorias de desempenho para carregar arquivos de tradução.
- Adicionada opção para desabilitar o gerenciamento de acesso no Weblate em favor do gerenciamento 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.

4.49.4 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 arquivo 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 arquivos XLIFF.
- Adicionado suporte para não agrupar textos longos em arquivos PO gettext.
- Adicionado botão para copiar o link permanente para a 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.

4.49.5 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.

4.49.6 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 grandes arquivos de tradução.
- Adicionado suporte para bloquear o registro de certos e-mails.
- Os usuários agora podem excluir seus 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.

4.49.7 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 usuário.
- Corrigida a pesquisa em componente /projeto/tido o site por traduções.

4.49.8 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 arquivos monolíngues de formato diferente.
- Melhoradas as mensagens de erro na autenticação com falha.
- Mantelém o estado da página ao remover a palavra do glossário.
- Adicionado link direto para editar a tradução do idioma secundário.
- Adicionada verificação de qualidade do formato Perl.
- Adicionado suporte para rejeitar senhas reutilizadas.
- Estendida a barra de ferramentas para edição de idiomas RTL.

4.49.9 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 senha não autentica mais o usuário.
- Corrigido um possível desvio de CAPTCHA em caso de senha esquecida.

4.49.10 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 senha agora é validada contra falhas comuns por padrão.
- Notificar os usuários sobre atividades importantes em suas contas, como alteração de senha.
- 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 registro de auditoria.
- Pede a confirmação da senha 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 os links de pesquisa podem ser compartilhados.
- Incluídas informações do texto fonte e capturas de tela na API.
- Permite sobrescrever traduções por meio de envio pela API.

4.49.11 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 arquivo atual ao baixar a tradução.
- Retorna HTTP 404 ao tentar acessar o projeto onde o usuário não tem privilégios.

4.49.12 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 usuário.
- Adiciona opção para construir listas de componentes automaticamente.
- O painel padrão para usuários 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 usando ACL de grupo.
- Adicionado controle de privilégios mais refinado.
- Várias pequenas melhorias na interface do usuário.

4.49.13 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 arquivos fonte no editor local.
- Adicionado suporte para configuração de teclado visual com caracteres especiais.
- Melhorado o gerenciamento de captura de tela 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 arquivo.

4.49.14 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 desenvolvedor.
- Melhor compatibilidade com várias versões git na extensão do exportador git.
- Incluído por projeto 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.

4.49.15 Weblate 2.10.1

Lançado em 20 de janeiro de 2017.

- Não vaza a existência da conta no formulário de redefinição de senha (CVE-2017-5537).

4.49.16 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 usuário 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.

4.49.17 Weblate 2.9

Lançado em 4 de novembro de 2016.

- Parâmetros estendidos para o comando de gerenciamento createadmin.
- Estendido import_json para ser capaz de lidar com componentes existentes.
- Adicionado suporte para arquivos 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.
- Registra exclusão de 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 aba de preferências selecionada após salvar.
- 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.

4.49.18 Weblate 2.8

Lançado em 31 de agosto de 2016.

- Melhorias na documentação.
- Traduções.
- Atualizadas as bibliotecas JavaScript empacotadas.
- Adicionado o comando de gerenciamento 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.

4.49.19 Weblate 2.7

Lançado em 10 de julho de 2016.

- Removida a tradução de máquina do Google web translate.
- 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 gerenciamento `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.

4.49.20 Weblate 2.6

Lançado em 28 de abril de 2016.

- Coeigida a validação de componentes com filtro de idioma.
- Melhorado suporte para arquivos 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.

4.49.21 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 gerenciamento para adicionar sugestões a partir da linha de comando.
- Adicionado suporte para mesclar comentários ao enviar arquivo.
- 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 em um 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 arquivos PHP.

- Melhorias de desempenho para verificações de qualidade.
- Corrigida a pesquisa para todo o site por verificações com falha.
- Adicionada opção para especificar o idioma fonte.
- Melhorado suporte para arquivos 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 gerenciamento para 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”).
- Esclarecida a terminologia sobre textos que necessitam ação e textos não traduzidas.
- Suporte para Python 3.
- Removido o suporte ao Django 1.7.
- Removida a dependência do msginit para a criação de novos arquivos 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 arquivos 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.

4.49.22 Weblate 2.4

Lançado em 20 de setembro de 2015.

- Melhorado o suporte para arquivos PHP.
- Capacidade de adicionar ACL a usuário 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, pode ser necessário atualizar 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 gerenciamento 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 gerenciar 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 arquivo.
- 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 arquivos CSV.
- Reescrito o tratamento de arquivos estáticos.
- Links diretos de login/registro para serviço de terceiros, se for o único.
- Faz commit de alterações pendentes na remoção da conta.
- Adiciona o comando de gerenciamento 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 arquivos para adicionar para commit.

4.49.23 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 arquivos de propriedade Java.
- Permita aceitar sugestão sem edição.
- Melhorado o suporte para Google OAuth 2.0.
- Adicionado suporte para arquivos .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 desabilitar a edição do modelo monolíngue.
- Visualização mais detalhada do status do repositório.
- Link direto 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.

4.49.24 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 arquivos de tradução.
- Adicionado suporte para traduções JSON monolíngues.
- Registra o bloqueio de componentes em um histórico.
- Suporte para edição de idioma fonte (modelo) para traduções monolíngues.
- Adicionado suporte básico para Gerrit.

4.49.25 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 arquivos.
- 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.

4.49.26 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 arquivo JSON.

- Corrigida a geração de arquivos mo em certos casos.
- Adicionado suporte a notificações do GitLab.
- Adicionado suporte a desabilitar sugestões de tradução.
- Suporte ao Django 1.7.
- Os projetos com ACL agora têm gerenciamento de usuários.
- 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.

4.50 Série 1.x do Weblate

4.50.1 Weblate 1.9

Lançado em 6 de maio de 2014.

- Compatibilidade com Django 1.6.
- Compatibilidade descontinuada com Django 1.4.
- Comandos de gerenciamento para bloquear/desbloquear traduções.
- Melhorado suporte para arquivos Qt TS.
- Os usuários agora podem excluir sua conta.
- Avatares podem ser desabilitados.
- 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.

4.50.2 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 usuários anônimos.
- Suporte a login usando serviços de terceiros, confira o manual para obter mais detalhes.
- Os usuários podem fazer login por e-mail em vez de nome de usuário.
- Melhorias na documentação.
- Melhorada a revisão de textos fontes.
- Pesquisa em todos os textos.

- Melhor rastreamento de textos fonte.
- Proteção por captcha para registro.

4.50.3 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.
- Usuários 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 registradas no histórico.
- Melhorias de desempenho para a visualização de tradução.

4.50.4 Weblate 1.6

Lançado em 25 de julho de 2013.

- Melhor tratamento de erros no registro.
- 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 reverter-lo.
- Adicionado suporte para API do Google Tradutor.
- Suporte adicional para gerenciamento 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.

4.50.5 Weblate 1.5

Lançado em 16 de abril de 2013.

- Confira o manual para as instruções de atualização.
- Adicionadas páginas públicas de usuários.
- 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 usuário.
- Opções estendidas de comando administrativo `import_project`.
- Compatível com Django 1.5.
- Avatares são agora mostrados usando libavatar.
- 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 arquivos adicionais.
- Reescrita a pesquisa para melhor desempenho e experiência do usuário.
- Nova interface para traduções de máquina.
- Adicionado suporte para arquivos po monolíngues.
- Estende a quantidade de metadados armazenados em cache para melhorar a velocidade de várias pesquisas.
- Agora mostra contagem de palavras também.

4.50.6 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 links diretos 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.

4.50.7 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.
- Melhorado o processamento de verificações 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 arquivo 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 usuários para traduções e textos fonte.
- Melhor rastreamento de logs de alteração.
- As alterações agora podem ser monitoradas usando RSS.
- Melhorado o suporte para idiomas RTL.

4.50.8 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 você estiver atualizando.
- Corrigidos pequenos problemas com repositórios git vinculados.
- Nova página de introdução para engajar as pessoas com tradução usando Weblate.
- Adicionados widgets que podem ser usados para promover projetos de tradução.
- Adicionada opção para redefinir o repositório para a origem (para usuários privilegiados).
- O projeto ou componente agora pode ser bloqueado para traduções.
- Possibilidade de desabilitar 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 arquivo `po`.
- Inclui algumas estatísticas na página Sobre.
- Adiciona suporte (e exige) `django-registration 0.8`.
- Fazer cache de contagens de textos contendo verificações com falha.
- Verificação de requisitos durante a configuração.
- Melhorias na documentação.

4.50.9 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.

4.50.10 Weblate 1.0

Lançado em 10 de maio de 2012.

- Melhorada a validação ao adicionar/salvar componente.
- Suporte experimental para arquivos de componentes do Android (precisa de `tkit` com patch aplicado).
- As atualizações dos ganchos são executadas em segundo plano.
- Melhoradas as instruções de instalação.
- Melhorada a navegação no dicionário.

4.51 Série 0.x do Weblate

4.51.1 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.
- Alterado o layout de arquivo.
- Atualiza scripts auxiliares para Django 1.4.
- Melhorada a navegação durante a tradução.
- Melhor tratamento de renomeações de arquivos 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.
- A verificação de textos não traduzidos ignora mensagens de texto apenas de formato.

4.51.2 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 gerenciamento agora também contém ações úteis para atualizar dados.
- Registra o log de alterações feitas pelos usuários.
- Capacidade de adiar o commit para Git para gerar menos commits de um único usuário.
- Possibilidade de navegar nas verificações com falha.
- Tradução automática usando 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.

4.51.3 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.

4.51.4 Weblate 0.6

Lançado em 14 de fevereiro de 2012.

- Adicionadas várias verificações para mensagens traduzidas.
- Controle de acesso ajustável.
- Melhorado o tratamento de traduções com novas linhas.
- Adicionada classificação de tabelas do lado do cliente.
- Verifique as instruções de atualização caso você esteja atualizando.

4.51.5 Weblate 0.5

Lançado em 12 de fevereiro de 2012.

- **Suporte para tradução de máquina usando os seguintes serviços online:**
 - Apertium
 - Microsoft Translator
 - MyMemory
- Várias novas traduções.
- 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 arquivos.
- Downloads de arquivos corrigidos ao usar FastCGI (e possivelmente outros).

4.51.6 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.

4.51.7 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.

4.51.8 Weblate 0.2

Lançado em 7 de fevereiro de 2012.

- Melhorada a validação de vários formulários.
- Avisa os usuários 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.

4.51.9 Weblate 0.1

Lançado em 6 de fevereiro de 2012.

- Lançamento inicial.

W

wlc, [160](#)

wlc.config, [161](#)

wlc.main, [161](#)

HTTP Routing Table

/	GET /api/components/(string:project)/(string:component)/(string:group)/, 135
ANY /, 107	GET /api/components/(string:project)/(string:component)/(string:group)/, 131
/api	GET /api/components/(string:project)/(string:component)/(string:group)/, 133
GET /api/, 109	GET /api/components/(string:project)/(string:component)/(string:group)/, 133
/api/addons	GET /api/components/(string:project)/(string:component)/(string:group)/, 132
GET /api/addons/, 147	GET /api/components/(string:project)/(string:component)/(string:group)/, 131
GET /api/addons/(int:id)/, 147	GET /api/components/(string:project)/(string:component)/(string:group)/, 135
PUT /api/addons/(int:id)/, 148	GET /api/components/(string:project)/(string:component)/(string:group)/, 133
DELETE /api/addons/(int:id)/, 148	GET /api/components/(string:project)/(string:component)/(string:group)/, 133
PATCH /api/addons/(int:id)/, 148	POST /api/components/(string:project)/(string:component)/(string:group)/, 148
/api/changes	POST /api/components/(string:project)/(string:component)/(string:group)/, 135
GET /api/changes/, 144	POST /api/components/(string:project)/(string:component)/(string:group)/, 131
GET /api/changes/(int:id)/, 144	POST /api/components/(string:project)/(string:component)/(string:group)/, 132
/api/component-lists	POST /api/components/(string:project)/(string:component)/(string:group)/, 134
GET /api/component-lists/, 148	PUT /api/components/(string:project)/(string:component)/(string:group)/, 130
GET /api/component-lists/(str:slug)/, 148	DELETE /api/components/(string:project)/(string:component)/(string:group)/, 130
POST /api/component-lists/(str:slug)/components/, 149	DELETE /api/components/(string:project)/(string:component)/(string:group)/, 136
PUT /api/component-lists/(str:slug)/, 149	PATCH /api/components/(string:project)/(string:component)/(string:group)/, 128
DELETE /api/component-lists/(str:slug)/, 149	
DELETE /api/component-lists/(str:slug)/components/(string:component)/(string:group)/, 149	
PATCH /api/component-lists/(str:slug)/, 149	
/api/components	/api/groups
GET /api/components/, 126	GET /api/groups/, 113
GET /api/components/(string:project)/(string:component)/(string:group)/, 126	GET /api/groups/(int:id)/, 114
GET /api/components/(string:project)/(string:component)/(string:group)/, 130	POST /api/groups/, 113
GET /api/components/(string:project)/(string:component)/(string:group)/, 130	POST /api/groups/(int:id)/, 115

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)/componentlist/, 116
 DELETE /api/groups/(int:id)/components/(int:component_id)/, 115
 DELETE /api/groups/(int:id)/languages/(string:language_code_id)/, 116
 DELETE /api/groups/(int:id)/projects/(string: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/, 122
 GET /api/projects/(string:project)/languages/, 125
 GET /api/projects/(string:project)/repositories/, 121
 GET /api/projects/(string:project)/statistics/, 126
 POST /api/projects/, 120
 POST /api/projects/(string:project)/components/, 123
 POST /api/projects/(string:project)/repositories/, 122

/api/roles

GET /api/roles/, 117
 POST /api/roles/, 117
 DELETE /api/roles/(int:id)/, 117
 PATCH /api/roles/(int:id)/, 117

/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_id)/, 147
 PATCH /api/screenshots/(int:id)/, 147

/api/tasks

GET /api/tasks/, 150
 GET /api/tasks/(str:uuid)/, 150

/api/translations

GET /api/translations/, 136
 GET /api/translations/(string:project)/(string:component)/, 136
 GET /api/translations/(string:project)/(string:component)/, 138
 GET /api/translations/(string:project)/(string:component)/, 140
 GET /api/translations/(string:project)/(string:component)/, 141
 GET /api/translations/(string:project)/(string:component)/, 141
 GET /api/translations/(string:project)/(string:component)/, 139
 POST /api/translations/(string:project)/(string:component)/, 139
 POST /api/translations/(string:project)/(string:component)/, 140
 POST /api/translations/(string:project)/(string:component)/, 141
 POST /api/translations/(string:project)/(string:component)/, 139

DELETE /api/translations/(string:project)/(string:component)/(string:language)/, 138
 POST /hooks/complaint/, 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)/, 110
 GET /api/users/(str:username)/notifications/, 112
 GET /api/users/(str:username)/notifications/(int:subscription_id)/, 112
 GET /api/users/(str:username)/statistics/, 112
 POST /api/users/, 110
 POST /api/users/(str:username)/groups/, 112
 POST /api/users/(str:username)/notifications/, 112
 PUT /api/users/(str:username)/, 111
 PUT /api/users/(str:username)/notifications/(int:subscription_id)/, 113
 DELETE /api/users/(str:username)/, 111
 DELETE /api/users/(str:username)/groups/, 112
 DELETE /api/users/(str:username)/notifications/(int:subscription_id)/, 113
 PATCH /api/users/(str:username)/, 111
 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/, 153
 POST /hooks/github/, 151

Símbolos

- .XML resource file
 - file format, [91](#)
- add
 - auto_translate opção de linha de comando, [388](#)
- addon
 - install_addon opção de linha de comando, [395](#)
- age
 - commit_pending opção de linha de comando, [389](#)
- author
 - add_suggestions opção de linha de comando, [388](#)
- author-email
 - wlc opção de linha de comando, [158](#)
- author-name
 - wlc opção de linha de comando, [158](#)
- base-file-template
 - import_project opção de linha de comando, [393](#)
- check
 - importusers opção de linha de comando, [394](#)
- config
 - wlc opção de linha de comando, [156](#)
- config-section
 - wlc opção de linha de comando, [156](#)
- configuration
 - install_addon opção de linha de comando, [395](#)
- convert
 - wlc opção de linha de comando, [157](#)
- email
 - createadmin opção de linha de comando, [390](#)
- file-format
 - import_project opção de linha de comando, [393](#)
- force
 - loadpo opção de linha de comando, [396](#)
- force-commit
 - pushgit opção de linha de comando, [397](#)
- format
 - wlc opção de linha de comando, [156](#)
- fuzzy
 - wlc opção de linha de comando, [158](#)
- ignore
 - import_json opção de linha de comando, [391](#)
- inconsistent
 - auto_translate opção de linha de comando, [388](#)
- input
 - wlc opção de linha de comando, [158](#)
- key
 - wlc opção de linha de comando, [156](#)
- lang
 - loadpo opção de linha de comando, [396](#)
- language-code
 - list_translators opção de linha de comando, [395](#)
- language-map
 - import_memory opção de linha de comando, [392](#)
- language-regex
 - import_project opção de linha de comando, [393](#)
- license
 - import_project opção de linha de comando, [393](#)
- license-url
 - import_project opção de linha de comando, [393](#)
- main-component
 - import_json opção de linha de comando, [391](#)
 - import_project opção de linha de comando, [393](#)
- method
 - wlc opção de linha de comando, [158](#)
- mode
 - auto_translate opção de linha de

comando, 389

--mt
auto_translate opção de linha de comando, 388

--name
createadmin opção de linha de comando, 390

--name-template
import_project opção de linha de comando, 393

--new-base-template
import_project opção de linha de comando, 393

--no-password
createadmin opção de linha de comando, 390

--no-privs-update
setupgroups opção de linha de comando, 397

--no-projects-update
setupgroups opção de linha de comando, 397

--no-update
setuplang opção de linha de comando, 398

--output
wlc opção de linha de comando, 158

--overwrite
auto_translate opção de linha de comando, 388
wlc opção de linha de comando, 158

--password
createadmin opção de linha de comando, 390

--project
import_json opção de linha de comando, 391

--source
auto_translate opção de linha de comando, 388

--threshold
auto_translate opção de linha de comando, 389

--update
createadmin opção de linha de comando, 390
import_json opção de linha de comando, 391
install_addon opção de linha de comando, 395

--url
wlc opção de linha de comando, 156

--user
auto_translate opção de linha de comando, 388

--username
createadmin opção de linha de comando, 390

--vcs
import_project opção de linha de comando, 393

A

add_suggestions
weblate admin command, 388

add_suggestions opção de linha de comando
--author, 388

ADMINS
setting, 211

AKISMET_API_KEY
setting, 345

ALLOWED_HOSTS
setting, 211

Android
file format, 85

ANONYMOUS_USER_NAME
setting, 345

API, 106, 155, 160

Apple strings
file format, 86

ARB
file format, 90

AUDITLOG_EXPIRY
setting, 346

AUTH_LOCK_ATTEMPTS
setting, 346

AUTH_TOKEN_VALID
setting, 347

auto_translate
weblate admin command, 388

auto_translate opção de linha de comando
--add, 388
--inconsistent, 388
--mode, 389
--mt, 388
--overwrite, 388
--source, 388
--threshold, 389
--user, 388

AUTO_UPDATE
setting, 346

AUTOFIX_LIST
setting, 347

AVATAR_URL_PREFIX
setting, 347

B

BACKGROUND_TASKS
setting, 348

BaseAddon (*classe em weblate.addons.base*), 433

BASIC_LANGUAGES
setting, 348

bilingual
translation, 77

BITBUCKETSERVER_CREDENTIALS
 setting, 356
BORG_EXTRA_ARGS
 setting, 348

C

CACHE_DIR
 setting, 349
can_install() (método de classe *weblate.addons.base.BaseAddon*), 433
CELERY_BACKUP_OPTIONS, 167, 184
CELERY_BEAT_OPTIONS, 167, 184
CELERY_MAIN_OPTIONS, 167, 184
CELERY_MEMORY_OPTIONS, 167, 184
CELERY_NOTIFY_OPTIONS, 167, 184
celery_queues
 weblate admin command, 389
CELERY_TRANSLATE_OPTIONS, 167, 184
changes
 wlc opção de linha de comando, 157
CHECK_LIST
 setting, 349
checkgit
 weblate admin command, 389
cleanup
 wlc opção de linha de comando, 157
cleanup_ssh_keys
 weblate admin command, 390
cleanuptrans
 weblate admin command, 390
Comma separated values
 file format, 92
Command (classe em *wlc.main*), 161
COMMENT_CLEANUP_DAYS
 setting, 350
commit
 wlc opção de linha de comando, 157
commit_pending
 weblate admin command, 389
commit_pending opção de linha de comando
 --age, 389
COMMIT_PENDING_HOURS
 setting, 350
commitgit
 weblate admin command, 389
configure() (método de classe *weblate.addons.base.BaseAddon*), 433
CONTACT_FORM
 setting, 350
createadmin
 weblate admin command, 390
createadmin opção de linha de comando
 --email, 390
 --name, 390
 --no-password, 390
 --password, 390
 --update, 390

 --username, 390
CSP_CONNECT_SRC
 setting, 349
CSP_FONT_SRC
 setting, 349
CSP_IMG_SRC
 setting, 349
CSP_SCRIPT_SRC
 setting, 349
CSP_STYLE_SRC
 setting, 349
CSV
 file format, 92

D

daily() (método de classe *weblate.addons.base.BaseAddon*), 433
DATA_DIR
 setting, 350
DATABASE_BACKUP
 setting, 351
DATABASES
 setting, 211
DEBUG
 setting, 211
DEFAULT_ACCESS_CONTROL
 setting, 351
DEFAULT_ADD_MESSAGE
 setting, 352
DEFAULT_ADDON_MESSAGE
 setting, 352
DEFAULT_ADDONS
 setting, 352
DEFAULT_AUTO_WATCH
 setting, 352
DEFAULT_COMMIT_MESSAGE
 setting, 352
DEFAULT_COMMITTER_EMAIL
 setting, 353
DEFAULT_COMMITTER_NAME
 setting, 353
DEFAULT_DELETE_MESSAGE
 setting, 352
DEFAULT_FROM_EMAIL
 setting, 211
DEFAULT_LANGUAGE
 setting, 353
DEFAULT_MERGE_MESSAGE
 setting, 352
DEFAULT_MERGE_STYLE
 setting, 353
DEFAULT_PAGE_LIMIT
 setting, 361
DEFAULT_PULL_MESSAGE
 setting, 354
DEFAULT_RESTRICTED_COMPONENT
 setting, 352
DEFAULT_SHARED_TM

setting, 354
 DEFAULT_TRANSLATION_PROPAGATION
 setting, 354
 download
 wlc opção de linha de comando, 157
 DTD
 file format, 94
 dump_memory
 weblate admin command, 391
 dumpuserdata
 weblate admin command, 391

E

ENABLE_AVATARS
 setting, 354
 ENABLE_HOOKS
 setting, 354
 ENABLE_HTTPS
 setting, 354
 ENABLE_SHARING
 setting, 355
 EXTRA_HTML_HEAD
 setting, 355

F

file format
 .XML resource file, 91
 Android, 85
 Apple strings, 86
 ARB, 90
 Comma separated values, 92
 CSV, 92
 DTD, 94
 gettext, 79
 go-i18n, 89
 gotext, 90
 GWT properties, 83
 i18next, 88
 INI translations, 84
 Java properties, 83
 Joomla translations, 85
 JSON, 87
 mi18n lang, 83
 PHP strings, 87
 PO, 79
 Qt, 85
 RC, 95
 ResourceDictionary, 92
 RESX, 91
 Ruby YAML, 93
 Ruby YAML Ain't Markup Language, 93
 string resources, 85
 TS, 85
 WPF, 92
 XLIFF, 81
 XML, 94
 YAML, 93
 YAML Ain't Markup Language, 93

G

get() (*método wlc.Weblate*), 160
 get_add_form() (*método de classe weblate.addons.base.BaseAddon*), 433
 GET_HELP_URL
 setting, 355
 get_settings_form() (*método weblate.addons.base.BaseAddon*), 433
 gettext
 file format, 79
 GITEA_CREDENTIALS
 setting, 355
 GITHUB_CREDENTIALS
 setting, 356
 GITLAB_CREDENTIALS
 setting, 356
 go-i18n
 file format, 89
 GOOGLE_ANALYTICS_ID
 setting, 356
 gotext
 file format, 90
 GWT properties
 file format, 83

H

HIDE_REPO_CREDENTIALS
 setting, 357
 HIDE_VERSION
 setting, 357

I

i18next
 file format, 88
 import_demo
 weblate admin command, 391
 import_json
 weblate admin command, 391
 import_json opção de linha de comando
 --ignore, 391
 --main-component, 391
 --project, 391
 --update, 391
 import_memory
 weblate admin command, 392
 import_memory opção de linha de comando
 --language-map, 392
 import_project
 weblate admin command, 392
 import_project opção de linha de comando
 --base-file-template, 393
 --file-format, 393
 --language-regex, 393
 --license, 393
 --license-url, 393
 --main-component, 393

- name-template, 393
 - new-base-template, 393
 - vcs, 393
- importuserdata
 - weblate admin command, 394
- importusers
 - weblate admin command, 394
- importusers opção de linha de comando
 - check, 394
- INI translations
 - file format, 84
- install_addon
 - weblate admin command, 395
- install_addon opção de linha de comando
 - addon, 395
 - configuration, 395
 - update, 395
- INTERLEDGER_PAYMENT_POINTERS
 - setting, 357
- iOS
 - translation, 86
- IP_BEHIND_REVERSE_PROXY
 - setting, 357
- IP_PROXY_HEADER
 - setting, 358
- IP_PROXY_OFFSET
 - setting, 358

J

- Java properties
 - file format, 83
- Joomla translations
 - file format, 85
- JSON
 - file format, 87

L

- LEGAL_TOS_DATE
 - setting, 358
- LEGAL_URL
 - setting, 358
- LICENSE_EXTRA
 - setting, 359
- LICENSE_FILTER
 - setting, 359
- LICENSE_REQUIRED
 - setting, 359
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH
 - setting, 360
- list_languages
 - weblate admin command, 395
- list_translators
 - weblate admin command, 395
- list_translators opção de linha de comando
 - language-code, 395
- list_versions

- weblate admin command, 396
- list-components
 - wlc opção de linha de comando, 156
- list-languages
 - wlc opção de linha de comando, 156
- list-projects
 - wlc opção de linha de comando, 156
- list-translations
 - wlc opção de linha de comando, 157
- load() (*método wlc.config.WeblateConfig*), 161
- loadpo
 - weblate admin command, 396
- loadpo opção de linha de comando
 - force, 396
 - lang, 396
- LOCALIZE_CDN_PATH
 - setting, 360
- LOCALIZE_CDN_URL
 - setting, 360
- lock
 - wlc opção de linha de comando, 157
- lock_translation
 - weblate admin command, 396
- lock-status
 - wlc opção de linha de comando, 157
- LOGIN_REQUIRED_URLS
 - setting, 360
- LOGIN_REQUIRED_URLS_EXCEPTIONS
 - setting, 360
- ls
 - wlc opção de linha de comando, 157

M

- main() (*no módulo wlc.main*), 161
- MATOMO_SITE_ID
 - setting, 361
- MATOMO_URL
 - setting, 361
- mi18n lang
 - file format, 83
- monolingual
 - translation, 77
- move_language
 - weblate admin command, 396
- módulo
 - wlc, 160
 - wlc.config, 161
 - wlc.main, 161

N

- NEARBY_MESSAGES
 - setting, 361

P

- PAGURE_CREDENTIALS
 - setting, 362
- PHP strings
 - file format, 87

PIWIK_SITE_ID
 setting, 361
 PIWIK_URL
 setting, 361
 PO
 file format, 79
 post() (*método wlc. Weblate*), 160
 post_add() (*método blate.addons.base.BaseAddon*), 433
 post_commit() (*método blate.addons.base.BaseAddon*), 433
 post_push() (*método blate.addons.base.BaseAddon*), 433
 post_update() (*método blate.addons.base.BaseAddon*), 433
 pre_commit() (*método blate.addons.base.BaseAddon*), 434
 pre_push() (*método blate.addons.base.BaseAddon*), 434
 pre_update() (*método blate.addons.base.BaseAddon*), 434
 PRIVACY_URL
 setting, 362
 PRIVATE_COMMIT_EMAIL_OPT_IN
 setting, 362
 PRIVATE_COMMIT_EMAIL_TEMPLATE
 setting, 362
 PROJECT_BACKUP_KEEP_COUNT
 setting, 363
 PROJECT_BACKUP_KEEP_DAYS
 setting, 363
 PROJECT_NAME_RESTRICT_RE
 setting, 363
 PROJECT_WEB_RESTRICT_RE
 setting, 363
 Propostas Estendidas Python
 PEP 484, 429, 444
 pull
 wlc opção de linha de comando, 157
 push
 wlc opção de linha de comando, 157
 pushgit
 weblate admin command, 397
 pushgit opção de linha de comando
 --force-commit, 397
 Python, 160

Q

Qt
 file format, 85

R

RATELIMIT_ATTEMPTS
 setting, 363
 RATELIMIT_LOCKOUT
 setting, 364
 RATELIMIT_WINDOW
 setting, 364

RC
 file format, 95
 REDIS_PASSWORD, 181
 register_command() (*no módulo wlc.main*), 161
 REGISTRATION_ALLOW_BACKENDS
 setting, 364
 REGISTRATION_CAPTCHA
 setting, 364
 REGISTRATION_EMAIL_MATCH
 setting, 365
 REGISTRATION_OPEN
 setting, 365
 REGISTRATION_REBIND
 setting, 365
 repo
 wlc opção de linha de comando, 157
 REPOSITORY_ALERT_THRESHOLD
 setting, 365
 REQUIRE_LOGIN
 setting, 366
 reset
 wlc opção de linha de comando, 157
 ResourceDictionary
 file format, 92
 REST, 106
 RESX
 file format, 91
 RFC
 RFC 5646, 76
 Ruby YAML
 file format, 93
 Ruby YAML Ain't Markup Language
 file format, 93

S

save_state() (*método blate.addons.base.BaseAddon*), 434 we-
 SECRET_KEY
 setting, 212
 SENTRY_DSN
 setting, 366
 SERVER_EMAIL
 setting, 212
 SESSION_COOKIE_AGE_AUTHENTICATED
 setting, 366
 SESSION_ENGINE
 setting, 211
 setting
 ADMINS, 211
 AKISMET_API_KEY, 345
 ALLOWED_HOSTS, 211
 ANONYMOUS_USER_NAME, 345
 AUDITLOG_EXPIRY, 346
 AUTH_LOCK_ATTEMPTS, 346
 AUTH_TOKEN_VALID, 347
 AUTO_UPDATE, 346
 AUTOFIX_LIST, 347
 AVATAR_URL_PREFIX, 347

- BACKGROUND_TASKS, 348
- BASIC_LANGUAGES, 348
- BITBUCKETSERVER_CREDENTIALS, 356
- BORG_EXTRA_ARGS, 348
- CACHE_DIR, 349
- CHECK_LIST, 349
- COMMENT_CLEANUP_DAYS, 350
- COMMIT_PENDING_HOURS, 350
- CONTACT_FORM, 350
- CSP_CONNECT_SRC, 349
- CSP_FONT_SRC, 349
- CSP_IMG_SRC, 349
- CSP_SCRIPT_SRC, 349
- CSP_STYLE_SRC, 349
- DATA_DIR, 350
- DATABASE_BACKUP, 351
- DATABASES, 211
- DEBUG, 211
- DEFAULT_ACCESS_CONTROL, 351
- DEFAULT_ADD_MESSAGE, 352
- DEFAULT_ADDON_MESSAGE, 352
- DEFAULT_ADDONS, 352
- DEFAULT_AUTO_WATCH, 352
- DEFAULT_COMMIT_MESSAGE, 352
- DEFAULT_COMMITER_EMAIL, 353
- DEFAULT_COMMITER_NAME, 353
- DEFAULT_DELETE_MESSAGE, 352
- DEFAULT_FROM_EMAIL, 211
- DEFAULT_LANGUAGE, 353
- DEFAULT_MERGE_MESSAGE, 352
- DEFAULT_MERGE_STYLE, 353
- DEFAULT_PAGE_LIMIT, 361
- DEFAULT_PULL_MESSAGE, 354
- DEFAULT_RESTRICTED_COMPONENT, 352
- DEFAULT_SHARED_TM, 354
- DEFAULT_TRANSLATION_PROPAGATION, 354
- ENABLE_AVATARS, 354
- ENABLE_HOOKS, 354
- ENABLE_HTTPS, 354
- ENABLE_SHARING, 355
- EXTRA_HTML_HEAD, 355
- GET_HELP_URL, 355
- GITEA_CREDENTIALS, 355
- GITHUB_CREDENTIALS, 356
- GITLAB_CREDENTIALS, 356
- GOOGLE_ANALYTICS_ID, 356
- HIDE_REPO_CREDENTIALS, 357
- HIDE_VERSION, 357
- INTERLEDGER_PAYMENT_POINTERS, 357
- IP_BEHIND_REVERSE_PROXY, 357
- IP_PROXY_HEADER, 358
- IP_PROXY_OFFSET, 358
- LEGAL_TOS_DATE, 358
- LEGAL_URL, 358
- LICENSE_EXTRA, 359
- LICENSE_FILTER, 359
- LICENSE_REQUIRED, 359
- LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 360
- LOCALIZE_CDN_PATH, 360
- LOCALIZE_CDN_URL, 360
- LOGIN_REQUIRED_URLS, 360
- LOGIN_REQUIRED_URLS_EXCEPTIONS, 360
- MATOMO_SITE_ID, 361
- MATOMO_URL, 361
- NEARBY_MESSAGES, 361
- PAGURE_CREDENTIALS, 362
- PIWIK_SITE_ID, 361
- PIWIK_URL, 361
- PRIVACY_URL, 362
- PRIVATE_COMMIT_EMAIL_OPT_IN, 362
- PRIVATE_COMMIT_EMAIL_TEMPLATE, 362
- PROJECT_BACKUP_KEEP_COUNT, 363
- PROJECT_BACKUP_KEEP_DAYS, 363
- PROJECT_NAME_RESTRICT_RE, 363
- PROJECT_WEB_RESTRICT_RE, 363
- RATELIMIT_ATTEMPTS, 363
- RATELIMIT_LOCKOUT, 364
- RATELIMIT_WINDOW, 364
- REGISTRATION_ALLOW_BACKENDS, 364
- REGISTRATION_CAPTCHA, 364
- REGISTRATION_EMAIL_MATCH, 365
- REGISTRATION_OPEN, 365
- REGISTRATION_REBIND, 365
- REPOSITORY_ALERT_THRESHOLD, 365
- REQUIRE_LOGIN, 366
- SECRET_KEY, 212
- SENTRY_DSN, 366
- SERVER_EMAIL, 212
- SESSION_COOKIE_AGE_AUTHENTICATED, 366
- SESSION_ENGINE, 211
- SIMPLIFY_LANGUAGES, 366
- SINGLE_PROJECT, 367
- SITE_DOMAIN, 366
- SITE_TITLE, 367
- SPECIAL_CHARS, 367
- SSH_EXTRA_ARGS, 367
- STATUS_URL, 368
- SUGGESTION_CLEANUP_DAYS, 368
- UPDATE_LANGUAGES, 368
- URL_PREFIX, 368
- VCS_API_DELAY, 369
- VCS_BACKENDS, 369
- VCS_CLONE_DEPTH, 369
- WEBLATE_ADDONS, 369
- WEBLATE_EXPORTERS, 370
- WEBLATE_FORMATS, 370
- WEBLATE_GPG_IDENTITY, 371
- WEBLATE_MACHINERY, 371
- WEBSITE_REQUIRED, 371
- setupgroups
 - weblate admin command, 397
- setupgroups opção de linha de comando
 - no-privs-update, 397

--no-projects-update, 397
 setuplang
 weblate admin command, 398
 setuplang opção de linha de comando
 --no-update, 398
 show
 wlc opção de linha de comando, 157
 SIMPLIFY_LANGUAGES
 setting, 366
 SINGLE_PROJECT
 setting, 367
 SITE_DOMAIN
 setting, 366
 SITE_TITLE
 setting, 367
 SPECIAL_CHARS
 setting, 367
 SSH_EXTRA_ARGS
 setting, 367
 stats
 wlc opção de linha de comando, 157
 STATUS_URL
 setting, 368
 store_post_load() (método *weblate.addons.base.BaseAddon*), 434
 string resources
 file format, 85
 SUGGESTION_CLEANUP_DAYS
 setting, 368

T

translation
 bilingual, 77
 iOS, 86
 monolingual, 77
 TS
 file format, 85

U

unit_pre_create() (método *weblate.addons.base.BaseAddon*), 434
 unlock
 wlc opção de linha de comando, 157
 unlock_translation
 weblate admin command, 397
 UPDATE_LANGUAGES
 setting, 368
 updatechecks
 weblate admin command, 398
 updategit
 weblate admin command, 398
 upload
 wlc opção de linha de comando, 158
 URL_PREFIX
 setting, 368

V

VCS_API_DELAY

 setting, 369
 VCS_BACKENDS
 setting, 369
 VCS_CLONE_DEPTH
 setting, 369
 version
 wlc opção de linha de comando, 156
 variável de ambiente
 CELERY_BACKUP_OPTIONS, 167, 184
 CELERY_BEAT_OPTIONS, 167, 184
 CELERY_MAIN_OPTIONS, 167, 184
 CELERY_MEMORY_OPTIONS, 167, 184
 CELERY_NOTIFY_OPTIONS, 167, 184
 CELERY_TRANSLATE_OPTIONS, 167, 184
 POSTGRES_ALTER_ROLE, 179
 POSTGRES_CONN_MAX_AGE, 179
 POSTGRES_DATABASE, 179
 POSTGRES_DISABLE_SERVER_SIDE_CURSORS, 180
 POSTGRES_HOST, 179
 POSTGRES_PASSWORD, 179
 POSTGRES_PASSWORD_FILE, 179
 POSTGRES_PORT, 179
 POSTGRES_SSL_MODE, 179
 POSTGRES_USER, 179
 REDIS_DB, 180
 REDIS_HOST, 180
 REDIS_PASSWORD, 180, 181
 REDIS_PASSWORD_FILE, 180
 REDIS_PORT, 180
 REDIS_TLS, 181
 REDIS_VERIFY_SSL, 181
 ROLLBAR_ENVIRONMENT, 183
 ROLLBAR_KEY, 183
 SENTRY_DSN, 183
 SENTRY_ENVIRONMENT, 183
 SOCIAL_AUTH_SLACK_SECRET, 178
 WEB_WORKERS, 167, 184
 WEBLATE_ADD_ADDONS, 184
 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, 405
 WEBLATE_ALLOWED_HOSTS, 169, 211, 215, 216, 367
 WEBLATE_API_RATELIMIT_ANON, 109, 173
 WEBLATE_API_RATELIMIT_USER, 109, 174
 WEBLATE_AUTH_LDAP_BIND_DN, 175
 WEBLATE_AUTH_LDAP_BIND_PASSWORD, 175
 WEBLATE_AUTH_LDAP_BIND_PASSWORD_FILE, 175

WEBLATE_AUTH_LDAP_CONNECTION_OPTION_REFERENCE, 175
 WEBLATE_AUTH_LDAP_SERVER_URI, 175
 WEBLATE_AUTH_LDAP_USER_ATTR_MAP, 175
 WEBLATE_AUTH_LDAP_USER_DN_TEMPLATE, 175
 WEBLATE_AUTH_LDAP_USER_SEARCH, 175
 WEBLATE_AUTH_LDAP_USER_SEARCH_FILTER, 175
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION, 175
 WEBLATE_AUTH_LDAP_USER_SEARCH_UNION_DELIMITER, 175
 WEBLATE_AUTO_UPDATE, 182
 WEBLATE_AVATAR_URL_PREFIX, 174
 WEBLATE_BASIC_LANGUAGES, 173
 WEBLATE_BITBUCKETSERVER_HOST, 172
 WEBLATE_BITBUCKETSERVER_TOKEN, 172
 WEBLATE_BITBUCKETSERVER_USERNAME, 172
 WEBLATE_BORG_EXTRA_ARGS, 174
 WEBLATE_CONTACT_FORM, 169
 WEBLATE_CORS_ALLOWED_ORIGINS, 174
 WEBLATE_CSP_CONNECT_SRC, 173
 WEBLATE_CSP_FONT_SRC, 173
 WEBLATE_CSP_IMG_SRC, 173
 WEBLATE_CSP_SCRIPT_SRC, 173
 WEBLATE_CSP_STYLE_SRC, 173
 WEBLATE_DATABASE_BACKUP, 180
 WEBLATE_DEBUG, 168, 235
 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, 182
 WEBLATE_EMAIL_HOST, 181
 WEBLATE_EMAIL_HOST_PASSWORD, 181, 182
 WEBLATE_EMAIL_HOST_PASSWORD_FILE, 182
 WEBLATE_EMAIL_HOST_USER, 181
 WEBLATE_EMAIL_PORT, 181, 182
 WEBLATE_EMAIL_USE_SSL, 181, 182
 WEBLATE_EMAIL_USE_TLS, 181, 182
 WEBLATE_ENABLE_AVATARS, 174
 WEBLATE_ENABLE_HOOKS, 174
 WEBLATE_ENABLE_HTTPS, 170, 246
 WEBLATE_ENABLE_SHARING, 174
 WEBLATE_EXTRA_HTML_HEAD, 174
 WEBLATE_GET_HELP_URL, 182
 WEBLATE_GITEA_HOST, 172
 WEBLATE_GITEA_TOKEN, 172
 WEBLATE_GITEA_USERNAME, 171
 WEBLATE_GITHUB_HOST, 171, 235
 WEBLATE_GITHUB_TOKEN, 171
 WEBLATE_GITHUB_USERNAME, 171
 WEBLATE_GITLAB_HOST, 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, 171
 WEBLATE_IP_PROXY_HEADER, 170
 WEBLATE_LEGAL_URL, 182
 WEBLATE_LICENSE_FILTER, 173
 WEBLATE_LICENSE_REQUIRED, 173
 WEBLATE_LIMIT_TRANSLATION_LENGTH_BY_SOURCE_LENGTH, 174
 WEBLATE_LOCALIZE_CDN_PATH, 183
 WEBLATE_LOCALIZE_CDN_URL, 183
 WEBLATE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 171
 WEBLATE_LOGLEVEL, 168
 WEBLATE_LOGLEVEL_DATABASE, 168
 WEBLATE_NO_EMAIL_AUTH, 179
 WEBLATE_PAGURE_HOST, 172
 WEBLATE_PAGURE_TOKEN, 172
 WEBLATE_PAGURE_USERNAME, 172
 WEBLATE_PRIVACY_URL, 182
 WEBLATE_PRIVATE_COMMIT_EMAIL_OPT_IN, 174
 WEBLATE_PRIVATE_COMMIT_EMAIL_TEMPLATE, 174
 WEBLATE_RATELIMIT_ATTEMPTS, 173, 407
 WEBLATE_RATELIMIT_LOCKOUT, 173
 WEBLATE_RATELIMIT_WINDOW, 173
 WEBLATE_REGISTRATION_ALLOW_BACKENDS, 169
 WEBLATE_REGISTRATION_OPEN, 169
 WEBLATE_REGISTRATION_REBIND, 169
 WEBLATE_REMOVE_ADDONS, 184
 WEBLATE_REMOVE_APPS, 183
 WEBLATE_REMOVE_AUTOFIX, 184
 WEBLATE_REMOVE_CHECK, 183
 WEBLATE_REMOVE_LOGIN_REQUIRED_URLS_EXCEPTIONS, 171
 WEBLATE_REQUIRE_LOGIN, 171, 366
 WEBLATE_SAML_IDP_ENTITY_ID, 179
 WEBLATE_SAML_IDP_IMAGE, 179
 WEBLATE_SAML_IDP_TITLE, 179
 WEBLATE_SAML_IDP_URL, 179
 WEBLATE_SAML_IDP_X509CERT, 179
 WEBLATE_SECURE_PROXY_SSL_HEADER, 170, 171
 WEBLATE_SERVER_EMAIL, 169
 WEBLATE_SERVICE, 167, 185

WEBLATE_SILENCED_SYSTEM_CHECKS,	177
173, 243	
WEBLATE_SIMPLIFY_LANGUAGES,	172
WEBLATE_SITE_DOMAIN,	168, 213, 232, 367
WEBLATE_SITE_TITLE,	168
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_KEY,	177
WEBLATE_SOCIAL_AUTH_AZUREAD_OAUTH2_SECRET,	177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_KEY,	177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_SECRET,	177
WEBLATE_SOCIAL_AUTH_AZUREAD_TENANT_OAUTH2_TEAM_ID,	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,	177
WEBLATE_SOCIAL_AUTH_FACEBOOK_SECRET,	177
WEBLATE_SOCIAL_AUTH_FEDORA,	178
WEBLATE_SOCIAL_AUTH_GITEA_API_URL,	177
WEBLATE_SOCIAL_AUTH_GITEA_KEY,	177
WEBLATE_SOCIAL_AUTH_GITEA_SECRET,	177
WEBLATE_SOCIAL_AUTH_GITHUB_KEY,	176
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_KEY,	176
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_NAME,	176
WEBLATE_SOCIAL_AUTH_GITHUB_ORG_SECRET,	176
WEBLATE_SOCIAL_AUTH_GITHUB_SECRET,	176
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_ID,	176
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_KEY,	176
WEBLATE_SOCIAL_AUTH_GITHUB_TEAM_SECRET,	176
WEBLATE_SOCIAL_AUTH_GITLAB_API_URL,	177
WEBLATE_SOCIAL_AUTH_GITLAB_KEY,	177
WEBLATE_SOCIAL_AUTH_GITLAB_SECRET,	177
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_KEY,	177
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET,	177
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_DOMAINS,	177
WEBLATE_SOCIAL_AUTH_GOOGLE_OAUTH2_WHITELISTED_EMAILS,	177
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ACCESS_TOKEN_URL,	178
WEBLATE_SOCIAL_AUTH_KEYCLOAK_ALGORITHM,	178
WEBLATE_SOCIAL_AUTH_KEYCLOAK_AUTHORIZATION_URL,	178
WEBLATE_SOCIAL_AUTH_KEYCLOAK_IMAGE,	178
WEBLATE_SOCIAL_AUTH_KEYCLOAK_KEY,	178
WEBLATE_SOCIAL_AUTH_KEYCLOAK_PUBLIC_KEY,	178
WEBLATE_SOCIAL_AUTH_KEYCLOAK_SECRET,	178
WEBLATE_SOCIAL_AUTH_KEYCLOAK_TITLE,	178
WEBLATE_SOCIAL_AUTH_OIDC_KEY,	178
WEBLATE_SOCIAL_AUTH_OIDC_OIDC_ENDPOINT,	178
WEBLATE_SOCIAL_AUTH_OIDC_SECRET,	178
WEBLATE_SOCIAL_AUTH_OIDC_USERNAME_KEY,	178
WEBLATE_SOCIAL_AUTH_OPENSUSE,	178
WEBLATE_SOCIAL_AUTH_SLACK_KEY,	178
WEBLATE_SOCIAL_AUTH_UBUNTU,	178
WEBLATE_SSH_EXTRA_ARGS,	174
WEBLATE_STATUS_URL,	182
WEBLATE_TIME_ZONE,	170
WEBLATE_URL_PREFIX,	173
WEBLATE_WEBSITE_REQUIRED,	173
WEBLATE_WORKERS,	167, 184
WL_BRANCH,	342
WL_COMPONENT_NAME,	342
WL_COMPONENT_SLUG,	342
WL_COMPONENT_URL,	343
WL_ENGAGE_URL,	343
WL_FILE_FORMAT,	342
WL_FILEMASK,	342
WL_LANGUAGE,	342
WL_NEW_BASE,	342
WL_PATH,	342
WL_PREVIOUS_HEAD,	342
WL_PROJECT_NAME,	343
WL_PROJECT_SLUG,	342
WL_REPO,	342
WL_TEMPLATE,	342
WL_VCS,	342

W

Weblate (classe em *wlc*), 160
 Weblate admin command
 add_suggestions, 388
 auto_translate, 388

- celery_queues, 389
 - checkgit, 389
 - cleanup_ssh_keys, 390
 - cleanuptrans, 390
 - commit_pending, 389
 - commitgit, 389
 - createadmin, 390
 - dump_memory, 391
 - dumpuserdata, 391
 - import_demo, 391
 - import_json, 391
 - import_memory, 392
 - import_project, 392
 - importuserdata, 394
 - importusers, 394
 - install_addon, 395
 - list_languages, 395
 - list_translators, 395
 - list_versions, 396
 - loadpo, 396
 - lock_translation, 396
 - move_language, 396
 - pushgit, 397
 - setupgroups, 397
 - setuplang, 398
 - unlock_translation, 397
 - updatechecks, 398
 - updategit, 398
 - WEBLATE_ADDONS
 - setting, 369
 - 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, 405
 - WEBLATE_ALLOWED_HOSTS, 211, 215, 216, 367
 - WEBLATE_API_RATELIMIT_ANON, 109
 - WEBLATE_API_RATELIMIT_USER, 109
 - WEBLATE_AUTH_LDAP_BIND_PASSWORD, 175
 - WEBLATE_DEBUG, 235
 - WEBLATE_EMAIL_HOST_PASSWORD, 182
 - WEBLATE_EMAIL_PORT, 181, 182
 - WEBLATE_EMAIL_USE_SSL, 181, 182
 - WEBLATE_EMAIL_USE_TLS, 181, 182
 - WEBLATE_ENABLE_HTTPS, 246
 - WEBLATE_EXPORTERS
 - setting, 370
 - WEBLATE_FORMATS
 - setting, 370
 - WEBLATE_GITHUB_HOST, 235
 - WEBLATE_GPG_IDENTITY
 - setting, 371
 - WEBLATE_LOCALIZE_CDN_PATH, 183
 - WEBLATE_MACHINERY
 - setting, 371
 - WEBLATE_RATELIMIT_ATTEMPTS, 407
 - WEBLATE_REQUIRE_LOGIN, 366
 - WEBLATE_SECURE_PROXY_SSL_HEADER, 170
 - WEBLATE_SERVICE, 167
 - WEBLATE_SILENCED_SYSTEM_CHECKS, 243
 - WEBLATE_SITE_DOMAIN, 213, 232, 367
 - WEBLATE_WORKERS, 167, 184
 - WeblateConfig (*classe em wlc.config*), 161
 - WeblateException, 160
 - WEBSITE_REQUIRED
 - setting, 371
 - wlc, 155
 - módulo, 160
 - wlc opção de linha de comando
 - author-email, 158
 - author-name, 158
 - config, 156
 - config-section, 156
 - convert, 157
 - format, 156
 - fuzzy, 158
 - input, 158
 - key, 156
 - method, 158
 - output, 158
 - overwrite, 158
 - url, 156
 - changes, 157
 - cleanup, 157
 - commit, 157
 - download, 157
 - list-components, 156
 - list-languages, 156
 - list-projects, 156
 - list-translations, 157
 - lock, 157
 - lock-status, 157
 - ls, 157
 - pull, 157
 - push, 157
 - repo, 157
 - reset, 157
 - show, 157
 - stats, 157
 - unlock, 157
 - upload, 158
 - version, 156
 - wlc.config
 - módulo, 161
 - wlc.main
 - módulo, 161
 - WPF
 - file format, 92
- ## X
- XLIFF
 - file format, 81
 - XML
 - file format, 94

Y

YAML

file format, [93](#)

YAML Ain't Markup Language

file format, [93](#)